

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～

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 博士(工学)

渡辺 政彦

1. 自動運転システム

ブレーキやステアリングなどいわゆる手足を電子的に制御してきた時代から、ドライバー（運転手）という頭脳の代わりをする時代になってきました。自動運転システムを作るという事は、熟練したドライバーの臨機応変な運転スキルを制御システムに組み込むということです。[1] 「モノ」を対象にする制御工学モデルに加えて、「ヒト」を対象にする人間工学・認知科学・情報工学のモデルが必要になります。

熟練ヒューマンオペレータを3つの振舞（行動）レベルに分類した認知学者ラスムッセンのSRK (Skill-Rule-Knowledge) モデルを図1に示します。[2]

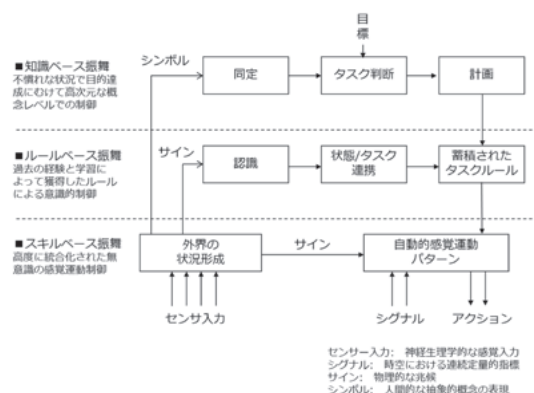


図1 熟練ヒューマンオペレータレベル

自動運転システムは「認知」「判断」「操作」という処理に分かれます。認知は歩行者や車両を的確に検出します。判断は認知の結果から実行すべき行動を決定します。操作は自車の走行を制御します。[3]

判断に関しては、今までの経験値から導出できるルールによる判断と、ルールでは想定していなかった状況での判断との2種類があります。内閣

府 SIP-adus (Automated Driving for Universal Services) [4]では自動走行システムを準自動走行システムと完全自動走行システムに分けています(表1)。SIP-adus レベル3は加速・操舵・制動を全て自動車がを行い、緊急時のみドライバーが対応する状態を準自動走行システムと定義しています。SIP-adus レベル4は加速・操舵・制動を全てドライバー以外が行い、ドライバーが全く関与しない状態を完全自動走行システムと定義しています。

表1 SIP-adusレベル

システム	レベル	特徴
完全自動走行システム	レベル4	加速・操舵・制動全てをドライバー(以外が実施。ドライバーが全く関与しない状態。 [2020年代後半])
高度運転支援システム (ADAS)	レベル3	準自動走行システム 加速・操舵・制動全てを自動車が実施。緊急時のみドライバーが対応する状態。DRMS (ドライバーレベルマネジメントシステム) [2020年代前半]
	レベル2	加速・操舵・制動を複数同時に自動車が行う状態。 ・自動緊急ブレーキ (AEB) ・高速道路での定速走行・車間距離制御 (ACC) ・車線逸脱防止システム (LKA) など 【一部実用化】[2017年以降]
安全運転支援システム (DSSS)	レベル1	一般道を中心に交通規制情報(信号・標識等)や見通しの悪い周辺の状況や、交通インフラからクルマに伝達し、ドライバーの安全運転を支援するシステム。具体的には、信号交差点の手前で赤信号情報を知らせることで赤信号の見落としを防止したり、見通しの悪い交差点で接近車両の情報を知らせることで出会い頭の事故を防止するなど安全運転を支援。2011年のサービス開始。【実用化】

レベル3はSRKモデルのルールベース振舞で、過去の経験と学習によって獲得したルールによる意識的制御です。ルールでは想定していなかった場合に、ドライバーに判断を委ねるのがレベル3です。レベル4はSRKモデルの知識ベース振舞で不慣れた状況で目的達成にむけて高次元な概念レベルでの制御です。ルールでは想定していなかった場合でも何らかの手段により自動運転できるのがレベル4になります。自動運転システムとSRKモデルとの関係を図2に示します。

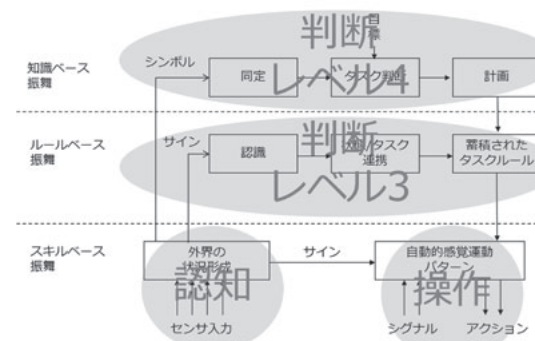


図2 自動運転システムとSRKモデルとの関係

2. ルールベースシステム

ルールベースシステムはエキスパートシステムまたはプロダクションシステムとも呼ばれ、if-then形式のルールを用いて問題解決を行うシステムのことです。古典的なルールベースではルール解析のオーバヘッドが問題になります。古典的なルールベースでは、既知の事実とルール群を順次照合し、適合するルールを実行していくため、処理性能は期待できません。

古典的なルールベースを改善するために、効率的にルールマッチングを行うReteアルゴリズムが提唱されました。Reteアルゴリズムは新たな事実が表明、更新されると、その事実に関連したルールだけを呼び出すようにして処理性能を上げます。[5,6] Reteアルゴリズムを実装したルールベースをプロダクションシステムと呼ぶことが多いです。プロダクションシステムの基本構成を図4に示します。[7]

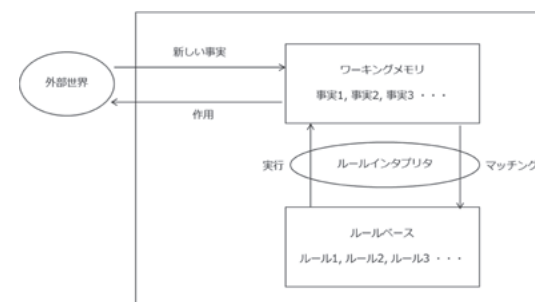


図4 プロダクションシステムの基本構成

ルールベースシステム（プロダクションシステム）は、エンタープライズ分野ではBRMS

(Business Rules Management System)として実務に適用されています。例えば、規則の組み合わせが複雑な保険の審査や保険料の算定、あるいは携帯電話の割引条件や料金算定など、ルール変更が頻繁な業務システムに使われています。[8]

古典的エキスパートシステムではルールの設定にはナレッジエンジニアと呼ばれる専門家が必要とされてきましたが、BRMSは決定表(デシジョンテーブル)によりプログラミングすることなしに、ルールの登録・変更ができ、一般の担当者でも運用できるようになっています。さらにBRMSでは、ビジネスルールをアプリケーションから切り離して管理することによってビジネス上の規則変更や価格改定に機敏に対応でき、かつ既存の会計処理システムとの独立性を高めたアーキテクチャにすることで、ルールの更新で全システムのリグレッションテスト工数を削減します。[9]

3. 自動運転システムへのルールベース適用の課題

ルールベースシステムを自動運転システムに適用する際の課題の1つに、リアルタイム性能があります。最近のBRMSでは、金融取引や、クレジットカード不正検知、ネットワークへのサイバーテロ検出など、リアルタイムで処理結果が得られなければ意味がないシステムにルールベースが適用されています。これを支える技術の1つにCEP(複合イベント処理)があります。[10]

CEPはある一定時間内で発生するイベントを複合的に解釈して、ある意味のある事実としてワーキングメモリに登録します(図5参照)。大量に発生し続けるイベント「センサ入力(神経生理学的な感覚入力)」、「シグナル(時空間における連続定量的指標)」を複合的に処理して、「サイン(物理的な兆候)」を抽出します(図1参照)。

ルールベースの振舞は、CEPが生成または抽出した意味あるイベント(サイン)を受け取り、「走る」「曲がる」「止まる」「つながる」それぞれのタスクの状態と調停し、必要なルールを決めます。ルールが成立すると操作系へのアクションを発火(実行)させます。

ルールベースとCEPを装備したシステムは、エンタープライズ分野ではエッジ側に配備され

ることが多いです。しかしながら、エンタープライズ分野のエッジと組み込み分野のエッジでは、使用できるCPU性能やメモリサイズなどのリソースが違います。例えば、エンタープライズ分野のエッジはパソコンですが、組み込み分野のエッジはECU (Electronic Control Unit) となります。エンタープライズ分野で良く用いられている Drools はルールベースと CEP の両方を装備したシステムです。Drools は JAVA 環境が動作するリッチなリソースを要求しますので、この点がリソース制約の厳しい組み込みエッジに実装する際の課題になります。

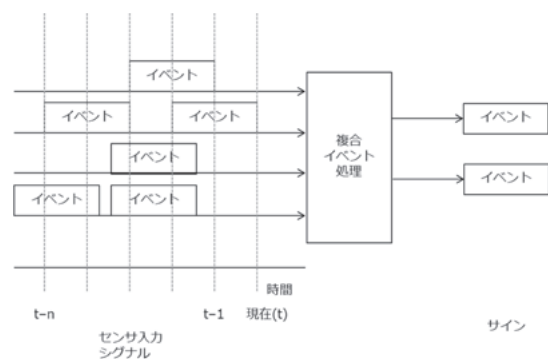


図5 複合イベント処理動作概要

もう1つの問題としてメモリ問題があります。BRMS に用いられるルールベースシステムには、OPS5、CLIPS、Drools など多くのものがあります。[11] これらのシステムではワーキングメモリの事実上の動的メモリを採用するため、メモリ枯渇やガベージコレクションが問題となります。この問題を解決するには静的メモリ構造を持つルールベースシステムが必要になります。

4. 自動運転システムのモデリング

自動運転システムでは「走る」「曲がる」「止まる」「つながる」を支える多くのタスクが存在し、ドライバーの状態、自車の状態、外部の状態（他車、天候、道路等々）などの状態が複雑化、巨大化しています。

自動運転システムで重要なことは、シグナルをサインに変える CEP 機能のための状態遷移モデル、多くのタスクと状態を調停する状態遷移モデル、そして熟練ドライバーのノウハウを蓄積した

ルール（決定表モデル）を開発、検証できるようにすることです（図6）。

操作系では制御工学のモデルベース開発を適用したように、判断系では情報工学のルールベース開発を適用したいものです。モデルベース開発ではモデルが資産であるように、ルールベース開発ではルールが資産になります。そして最新のルールベース開発環境では、ルールをモデルベースで開発できます。

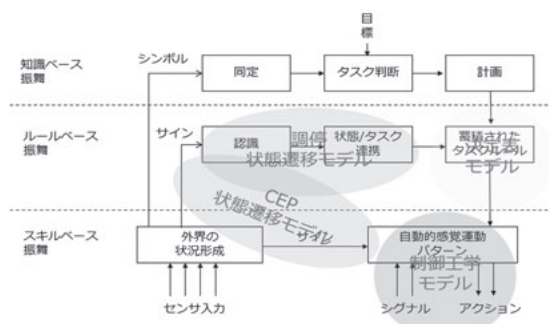


図6 自動運転システムモデル

情報工学（ソフトウェア工学）で培った標準形式（表記法）を活用して、ルールを蓄積します。状態遷移図・表は、ISO/IEC 11411 で定義されています。決定表は、ISO5086 で定義されています。詳しくは補足を参照ください。

5. 課題を解決する ZIPC R&B

SRK モデルと ZIPC Designer と ZIPC R&B の関係を図7に示します。各種自動走行に関する実験データからルール導出するために、状態遷移表による CEP モデリングと決定表によるルールモデリングを行います。ZIPC R&B はルールベースエンジンで、ファクト（事実）がワーキングメモリに登録されると、ファクトに関するルールを呼び出し、ルール条件すべてが真となると、アクション（作用）を実行します。

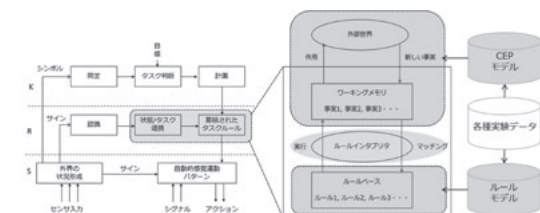


図7 SRKモデルとZIPC Designer/R&Bの関係

ZIPC Designer と ZIPC R&B の関係を図8に示します。ZIPC Designer は状態遷移表、状態遷移図を用いて CEP モデル、そして決定表によるルールモデルをエディットすることができます。ZIPC Designer は CEP モデルとルールモデルから C 言語コードを自動生成します。ZIPC Designer は ZIPC R&B に対応しない C 言語コードと対応するコードの両方を自動生成することができます。ZIPC R&B に対応する場合と対応しない場合の違いを知識表現の比較のところでも述べます。

PC 上で動作するルールベースエンジン ZIPC R&B は、ZIPC Designer ルールのツールチェーン機構を使うことで、他のツールと連携できます。制御工学モデルである MATLAB/Simulink モデルや、HUD (Head Up Display)、メータやセンターディスプレイなどの e コックピットをモデリングした HMI (Human Machine Interface) モデルなどと、ルールベースエンジンを連携して PC 上で動作することができます。

ZIPC Designer と ZIPC R&B は C 言語に対応で、かつ OS や RTOS には依存しないので、多くのターゲットへの移植が比較的容易で組み込みシステムに適しています。

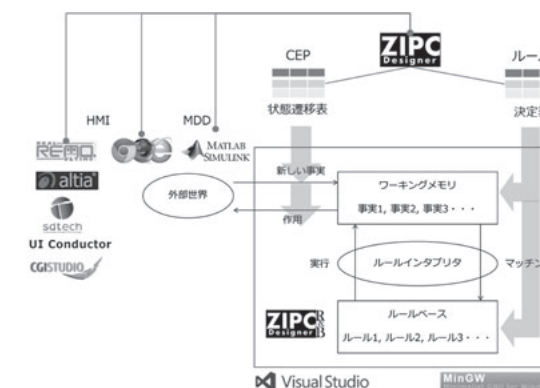


図8 ZIPC Designer と ZIPC R&B の関係

ZIPC Designer だけを用いてルールモデリングを行った場合は、知識をプログラムとして表現することになります。ZIPC R&B を用いた場合は、プロダクションシステムとして知識をデータとして表現することになります（図9）。

プログラムでは与えられた問題をどのように解

くかという知識をアルゴリズムとして直接プログラム中に埋め込むため、実行の順序を注意深くプログラミングできます。しかし知識を更新することがプログラムの構造の変化につながり、更新に多くの工数が必要になります。

プロダクションシステムでは、推論機構（ルールエンジン）と知識が分離されているので、推論機構の役割は手続き的表現と比べて単純になり、知識の更新が容易で人間にも理解しやすくなります。しかしながら、当然ですが推論機構が別途必要になります。[12]

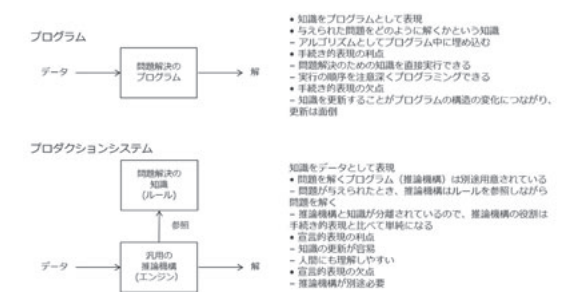


図9 知識表現の比較

3章の自動運転システムへのルールベース適用の課題で述べましたが、ルールベースのツールは OPS5、CLIPS、Drools など多くのものがあります。ZIPC Designer / R&B の特徴は以下になります。

- (1) ルールをモデリングするためのデザインテーブルを持っている
- (2) ある一定時間内での変化を処理するための CEP（複合イベント処理）をモデリングする状態遷移モデルを持っている。
- (3) 小規模リソースであるエッジ機器やリアルタイム性能が必要となる ECU などに搭載できるように、C 言語で書かれたリアルタイム性能を持つルールベースエンジンを持つ。
- (4) 静的メモリアロケーションアーキテクチャにより、メモリ資源枯渇問題やガベージコレクション問題を起こすことがない。

ZIPC Designer / R&B の特徴を示します(表2)。

表2 ルールベースツール比較表

製品名	デザインテーブル	CEP	C言語	メモリ配置
ZIPC R&B	○	○	○	静的
Drools(OSS/redhat)	○	○	×	動的
CLIPS(OSS)	×	×	○	動的
OPSS(OSS)	×	×	×	動的

6. 自動運転ルールベースシステムアーキテクチャ

自動運転システムを実現するためのSRKモデルとルールベースを組合せたアーキテクチャについて考察します。図1に示したSRKモデルから知識ベース振舞を除外し、さらにルールベース振舞から「状態/タスク連携」と「蓄積されたタスクルール」を除外したとします。そこに金沢大学で進めている自動運転のアーキテクチャ [13] を重ねたアーキテクチャを図10に示します。図10はルールのないルールベース振舞とスキルベース振舞から構成されています。ADAS（高度運転支援システム）をルールベースなしで構築するアーキテクチャ（ルールなしSRモデル）です。

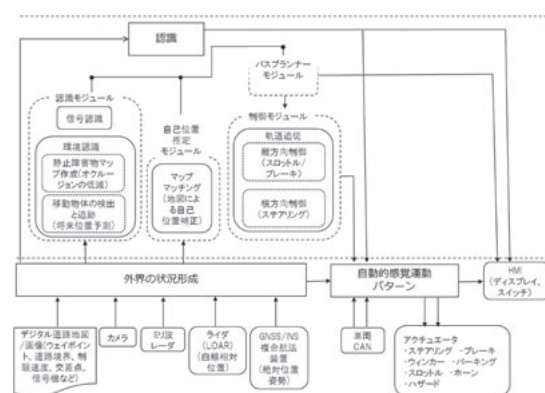


図10 ルールなしSRモデル

ルールなしSRモデルにルールを加えることで、ルールがある場合との構造の違いを明確にします。ルールありSRモデルが図11です。ルールありSRモデルでは、「認識」部の「認識モジュール」、「自己位置推定モジュール」、そして「パスプランナー」からのイベントを「状態/タスク連携」の「複合イベント処理 (CEPモデル)」が受信します。「複合イベント処理 (CEPモデル)」は、各イベントからルールに関係する事実 (事実) をワーキングメモリ上に生成、更新、削除します。事実とリンクされたルールがルールインタプリタによって呼び出され、ルールの条件が満たされるとアクションが作用します。この作用は「パスプランナーモジュール」に対して行われ、「パスプランナーモジュール」は「蓄積されたタスクルール」から過去の経験と学習によって獲得され

た意識的制御をフィードバックして、最終的な制御モジュールへのパラメータ値を決定します。

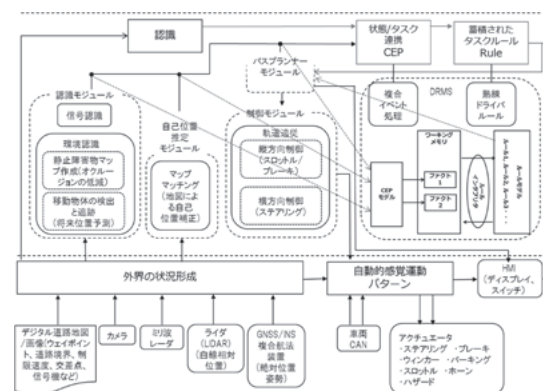


図11 ルールありSRモデル

SIP-adus レベル3は段階的に実現されていくと思われます。先読み運転知能を備えた熟練ドライバーモデルにより、ドライバーに早めにヒヤリハットの警告を与えることで予防安全を図ることができます。科学技術振興機構 (JST)「戦略的イノベーション創出推進プログラム」研究開発テーマ「高齢社会を豊かにする科学・技術・システムの創生」において、トヨタ自動車の井上秀雄氏は以下のように述べています。

「対歩行者・自転車ヒヤリハットシーンを分析した結果、従来の衝突回避システムの仕様では衝突を十分に回避できないシーンが存在することがわかった。特に遮蔽物から歩行者・自転車が出現する場合には、現行の衝突回避システムでは限界があり、今後はさらに潜在リスクを予想し、熟練ドライバーの様な運転知能を組込む必要がある。」 [24]

公益社団法人自動車技術会 (JSAE) から東京農工大学に管理運営権が移行したドライブレコーダを用いた収集したヒヤリハットのデータは2015年12月末で109,756件の登録があります。 [14、15]

一般社団法人日本自動車連盟 (JAF) は、危険予知・事故回避を目的とした「このとき、あなたは何に注意しますか」といった様々なシーンを提供し、事前の危険察知を行うドライバーの知見を提供します。 [16]

一般財団法人日本自動車研究所 (JARI) は、「JARI-ARV (拡張現実実験車)」を用いて実際の交通事故/危険場面を再現したテストコース走行実

験により、ヒューマンエラーを中心とする事故発生要因を分析や、事故や危険状況に至る際の走行データ (運転行動や車両挙動) を分析してドライバー特性を把握することで、適切な運転支援方法のあり方を検討すると共に、警報提供等による衝突回避効果の検討に活用を期待しています。 [17]

我が国にはこのような自動車を運転する場合のヒヤリハットに関する多くのデータがあります。こうした映像化や分析されたヒヤリハットのデータは、主にドライバーのトレーニングに用いられています。ルールベースシステムではこうしたヒヤリハットのデータを過去の経験と学習によって獲得したルールによる意識的制御として自動運転システムに組み込むことができます。

ルールベースシステムと既存システムは疎結合であり、このためルール更新が既存システムに影響を与えない構造のため、機能評価のためのリグレッションテスト工数は基本的には不要です。このようなアーキテクチャはエンタープライズ分野ではBRMS (Business Rules Management System) として既に実用化されています。ルールベースシステムを熟練ドライバーのルールを適用したDRMS (Driver Rules Management System) として導入することで、既存機能であるアンチロック・ブレーキ・システム (Antilock Brake System、略称: ABS) や横滑り防止装置 (Electronic Stability Control、略称 ESC) などをマイクロサービスとして呼び出すことができます。DRMSにより、ABSやESCなどの既存資産を上手く活用したレトロフィットを行うことができるのです。

ルールベースはルールに基づいて行動するシステムです。通信断絶などを考えると膨大なルールをクラウドではなくエッジ側に搭載する必要があります。このためにはメモリ制約を効率化し、リアルタイム性能を満足することが重要です。深層学習ではGPGPU (General-purpose computing on graphics processing units) といったリッチなデバイスを必要としますが、最適化されたルールベースエンジンを搭載したシステムではそのようなハイパフォーマンスデバイスは必要としません。ZIPC R&Bは非常に効率的なリソース運用をするルールベースエンジンです。

ルールベースはルールに基づくため、ルール以外の場面、いわゆる想定外の場面が起こった時に処理できないという指摘があります。SIP-adus レベル3では緊急時はドライバーに運転制御を移すことで想定外を処理します。このため、ドライバーが全く関与しないSIP-adus レベル4にはルールベースは適用できないという意見があります。しかしながら、自動車を運転することの目標は安全に目的地に到達することです。どんなに正確に早く目的地に到着しても、事故を起こしては元も子もありません。これは我々人間が運転している場合と全く変わらないことです。ルールのない想定外が発生した場合は、安全に停止することが自動運転システムには求められている目標です。こう考えると今はやりの深層学習は実はSRKモデルにおいてはスキルベース振舞における外界の状況形成において画像認識を行う部分に適用することが良いのかもしれませんが、想定外のシーンで深層学習が考えた行動で事故が発生した場合、ブラックボックスである深層学習では説明責任を果たすことはできないと思われます。「自動運転により運転者の責任が軽くなる分、メーカーの責任は重くなる。メーカー側の製造責任が問われることもありうる」のです。 [18]

想定外のシーンで安全に停止する処理に入ったときの各種データをクラウドにあげて、クラウド側でビッグデータ解析を行い、新たなルールを生成して、ルールを更新することになります。現在大量にある走行データはルールとすることでアセット (資産) 化することができるのです。

7. ルールベースプロセス

まずは御社のシステムのどの部分がルール化できるか検討しませんか？

8. ルールベースサンプル

そして、そのルールをモデル化して、機能とルールを分離したアーキテクチャであるプロダクションシステムにすることを検討してみませんか？

ご興味ある方、下記にお問い合わせください
宛先: キャッツ株式会社 AIセンター 担当者
Email: info@zipc.com

●トピック

■人工知能分類

人工知能を特徴別に4つに分類したものが表3です。[26]

表3 人工知能分類

レベル	特徴	実現方式
4	変数を発見する	深層学習 GPGPU
3	変数の重みを学習する	機械学習 (ビッグデータ解析) クラウドコンピューティング
2	探索や知識を使って、言われた通りにやる	プロダクションシステム エッジルールベースシステム
1	言われた通りにやる	プログラム マイコンで組み込みソフト

レベル1は言われた通りに行うもので、プログラムで実現します。電子立国日本と言われた時代、マイコンジャー炊飯器には名前の通り、マイコンを搭載し組み込みソフトウェアで米を炊く職人技を自動化しました。

レベル2では、探索や知識を用いて言われた通りに行います。レベル2はルールベースシステム (プロダクションシステム) として実現されます。レベル1とレベル2の違いは、5章の図9を参照してください。

レベル3は機械学習、レベル4は深層学習です。機械学習と深層学習の違いを表4に示します。[26,27]

表4 機械学習と深層学習の違い

学習方式 (レベル)	代表例	特徴
機械学習 (3)	小売店の来店客数予測。 来店客数に影響を及ぼす要因(変数)として、「曜日」「時間帯」「天候」「価格」などが考えられる。これらの変数が来店客数に与える影響の重みを算出し、それぞれの重みを考慮して来店客数を予測する。	数値予測など定量的な分析が得意
深層学習 (4)	画像認識。 人間は、経験的にイヌの写真とネコの写真を見分けることはできますが、イヌの写真とネコの写真の違いを決定づける「変数」が何であるかは分からない。ディープラーニングでは、この「変数」を見つけ出し、見つけ出した変数を用いて、イヌの写真とネコの写真を判別する。 機械学習で画像認識を行う場合、特徴(変数)を抽出していくことが難しい。実際に人が、猫とトラの子供を識別する特徴を知ることは難しい。ディープラーニングでは、非常に細かい部分まで特徴を抽出できるため、画像認識や音声認識の分野で幅広く活用されるようになった。 特徴(変数)の抽出にスパースコーディング理論を用いる。	画像認識やテキストマイニングなど定性的な分析が得意

機械学習は変数 (要因) が明確で、目的変数と説明変数に分けることができます。目的変数は予測したい変数で、表4の例では「来店客数」が目的変数となります。説明変数は目的変数を説明する変数で、「曜日」、「時間帯」、「天候」、「価格」が説明変数になります。変数の重みが日々蓄積されるデータによって更新されます。

深層学習は、説明変数が明確でない、つまり判断の手続きが明確に分からない場合でも、投入されたデータから変数 (特徴) を抽出して、解を出します。ルールベースと深層学習の違いを比較してそれぞれの特徴を明確にします (図12)。[28]

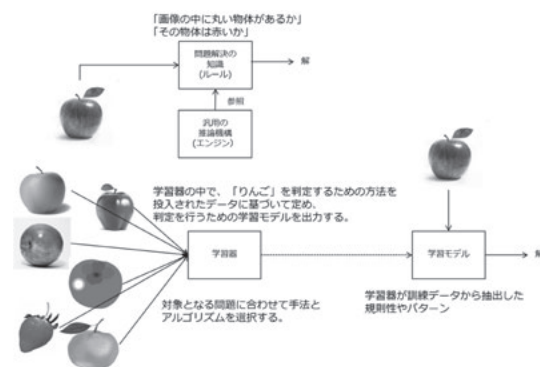


図12 ルールベースと深層学習の違い

ルールベースは、事務作業のような手続きが明確になっている処理に対して効率的かつ正確に機能します。ルールベースで「りんご」であるかどうかを特定するには、「画像の中に丸い物体があるか」、「その物体は赤いか」といったルールに基づいて判断することになります。青いりんごはどうか、赤く熟れた柿との区別はできるのかなどなど、人間がどのようにして「りんご」を認識しているかが完全に解明されていないので、ルールは発散しがちです。

深層学習では、判断の手続きが明確に分からなくても、学習モデルを利用して解を出します。学習モデルは大量のデータを対象となる問題に最適と思われるアルゴリズムを用いて学習器により生成されます。画像認識の場合、スパースコーディング理論が用いられます。スパースコーディング理論は1996年、OlshausenとFieldが脳視覚野の情報処理をモデル化し、画像がまばら (スパース) でも効率的に符号化 (コーディング) する仕組みです。

■自動運転システムにおける深層学習と

ルールベース

カメラなどの環境センサを用いて物体検出し、外界の状況を形成するには「深層学習 (ディープラーニング)」が用いられます。深層学習は画像

認識を得意としており、約1億2000万個の物体の画像情報をプログラミングなしにコンピュータが学習することで、自動的に物体を識別します。エヌビディア社はGPU (Graphic Processing Unit) を用いたNVIDIA DRIVE PX2を開発し1秒間に8兆回計算できる深層学習の環境を提供し、学習にかかる時間を以前の1/20~1/40に短縮したそうです。[19]

判断レベル4 (図2) では世界中の多様な走行条件でどこまでも走り抜けるためにあらゆる課題を解決しなければなりません。ルールベースは判断レベル3には有効でも、レベル4への適用には向きません。そこで、打開策となり得るのが深層学習だと言われています。[20]

深層学習を用いたGoogle社のGoogle DeepMindが囲碁で人間に4勝1敗で圧勝しましたが2つの弱点を露呈されました。1点目はAI (Artificial Intelligence: 人工知能) が明らかに誤りと思える判断を出力した場合にも、その原因の解析が極めて困難であることで、2点目はAIの結果的に正しい判断が人間には全く理解できない場合があることが指摘されています。[21] マイクロソフト社のAIであるTayが不適切な発言を繰り返して、ツイート開始から1日も経たないうちにつぶやきを停止したそうです。完全自動走行システムである判断レベル4への深層学習の適用にはまだまだ課題が残っているように思えます。

トヨタ自動車 AI 研究子会社 Toyota Research Institute 社 CEO Gill Pratt氏は以下のような発言をしています。

「強化学習は効果が非常に高いので、自動運転車の技術の選択肢から外すことはない。ただし、(予想できない挙動を示すといった懸念を払拭するために) 恐らくルールベースのAIと組み合わせることになるだろう」

深層学習で作成したニューラルネットワークは内部がブラックボックスに近く、自動運転車など命に関わるシステムの制御を完全に任せるには不安が残るということだと思われます。[22]

■ルールベースの適用範囲は広い

制御システムの進化は「知的化」することです。個々の単機能が複合的に融合することで知能化し

ています。自動運転システムはその最たるものですが、個別の機能も実は知能化が進んでいます。

複数のディスプレイが存在する場合、何時、何処に、どのように表示するかが大きな課題となり、さらに近年のADAS系機能表示を提供し、複数機能が同時に実行されている場合、優先順位と表示場所の考え方はより複雑になると指摘されています。[23]

戦略的イノベーション創造プログラム (SIP) 自動走行システム研究開発計画では、ドライバーと自動走行システムのHMI (Human Machine Interface) 技術の開発の重要性を以下のように指摘します。

「従来から実用化されている運転支援システムでは、安全な運転をすべき主体者はドライバーであり、システムはドライバーを側面から支援するものであったが、自動走行システムでは人とシステムが協調・一体となって走行の安全を確保することが求められる。そのためには、ドライバーと自動走行システムの関係を適正に保つための役割を定義し、意思疎通するためのインターフェースの開発が重要となる。」[25]

複雑化するHMI仕様を従来通りのソフトウェア開発で、要件、仕様、コーディング、テストを行っている、膨大な工数となります。HMI仕様の中で、表示するためのルールと表示するための機能を分離し、プロダクションシステムアーキテクチャを導入することがその解決方法の1つです。HMIへ表示するルールを暗黙知から形式知化することが求められています。

エンタープライズ業界では既にBRMS (Business Rule Management System) として実用され、効果を上げている。BRMSの特徴は、ルールと業務ロジックを分離することで、開発や保守の品質、費用、期間を向上させることができます (図13)。[29, 30]

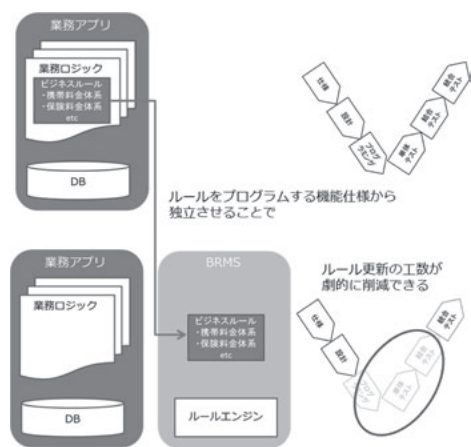


図13 BRMSの考え方

ルール更新が頻繁にあるような組み込みシステムにはBRMSの考え方を導入することは有効です。今まで、プログラムコードを生成することを前提にした機能仕様書から、ルールを抽出します。そしてルールを決定表でモデリングします。既存システムもしくは機能実現システムとのインターフェースを状態遷移表でモデリングします。ルールはルールエンジン上で実行します(図14)。

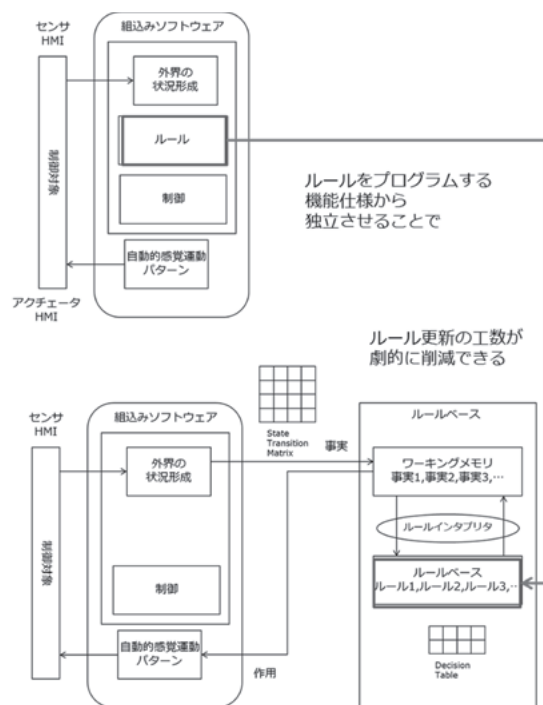


図14 組み込みシステムへのルールベース適用

このようなルールベースシステム(プロダクションシステムとも言う)を実現することで、ADASのような新しい分野で、ルール仕様が変わるシステムへの適用に効果があります。新しい分野ではルールが大きく進歩します。この進化の対応を従来のように、ソフトウェア仕様書をおこして、コーディングし、テストしていたら工数がかかりすぎてしまいます。

また、ルール更新作業は、ルールベースでは決定表を変更するだけなので、すぐに対応できます。一方で深層学習ではルールの変更をすぐに対応することは困難だと思われます。こうした点はルールベースが深層学習に比べて優位な点です。

3.3兆円でソフトバンクグループがARMを買収したニュースには驚きました。さて、そのARMの資料によると、マイコンやエッジ機器の市場を「エンベデッドインテリジェンス」市場と呼んでいるのです。そしてその規模は、2020年に300億ドル(3兆円)となり、モバイルアプリケーションプロセッサの市場規模である250億ドル(2.5兆円)を上回る予測をしています。この予測からもエッジ側の知能化はこれからますます増えると思われまます。

■ SRKモデルとサブサンプション・アーキテクチャ [31]

ウィキペディアのサブサンプション・アーキテクチャについての記述は、SRKモデルがサブサンプション・アーキテクチャであることを上手く説明しているのので、下記に引用します。

『サブサンプション・アーキテクチャ(Subsumption Architecture、包摂アーキテクチャ)とは、振舞に基づくロボット工学を起源とする人工知能の概念である。この用語は1986年にロドニー・ブルックスらが作った。

自律型ロボットやリアルタイムAIに幅広い影響を及ぼした。サブサンプション・アーキテクチャ(以下、SA)は、複雑な知的振舞を多数の「単純」な振舞モジュールに分割し、振舞のモジュールの階層構造を構築する。

各層は何らかの目的に沿った実装であり、上位層に行くに従ってより抽象的になる。

例えば、ロボットが持つ最下位層として「物体

を避ける」という振舞があり、その上位層として「うろつきまわる」という振舞があり、そのさらに上位に「世界を探索する」という振舞いがある。このような階層の最上位にあるのはそのロボットの究極の目的である「地図を作成する」という振舞かもしれない。各層は全センサーのデータを参照でき、アクチュエータを操作できる。ただし、他のタスクが入力を横取りしたり、出力を上書きしたりすることもある。この場合、下位層は高速な適応機構(反射)に相当し、上位層は本来の目的に沿って行動を導く制御を行う。』

6章の図11に示したルールありSRモデルとSAの関係を図15に示します。下位層は「ぶつからずに安全に動く」ための制御工学モデルであり、中間層は「目的地に最適経路で到着」するための制御工学モデルです。上位層は「安心して快適に」運転する情報工学モデルになります。中間層へのイベントを上位層が横取りし、パスプランナーモジュールに対して蓄積されたルール上の重みを上書きしたりすることもあります。この場合、下位層は高速な適応機構(反射)相当し、上位層は本来の目的に沿って行動を導く制御を行います。

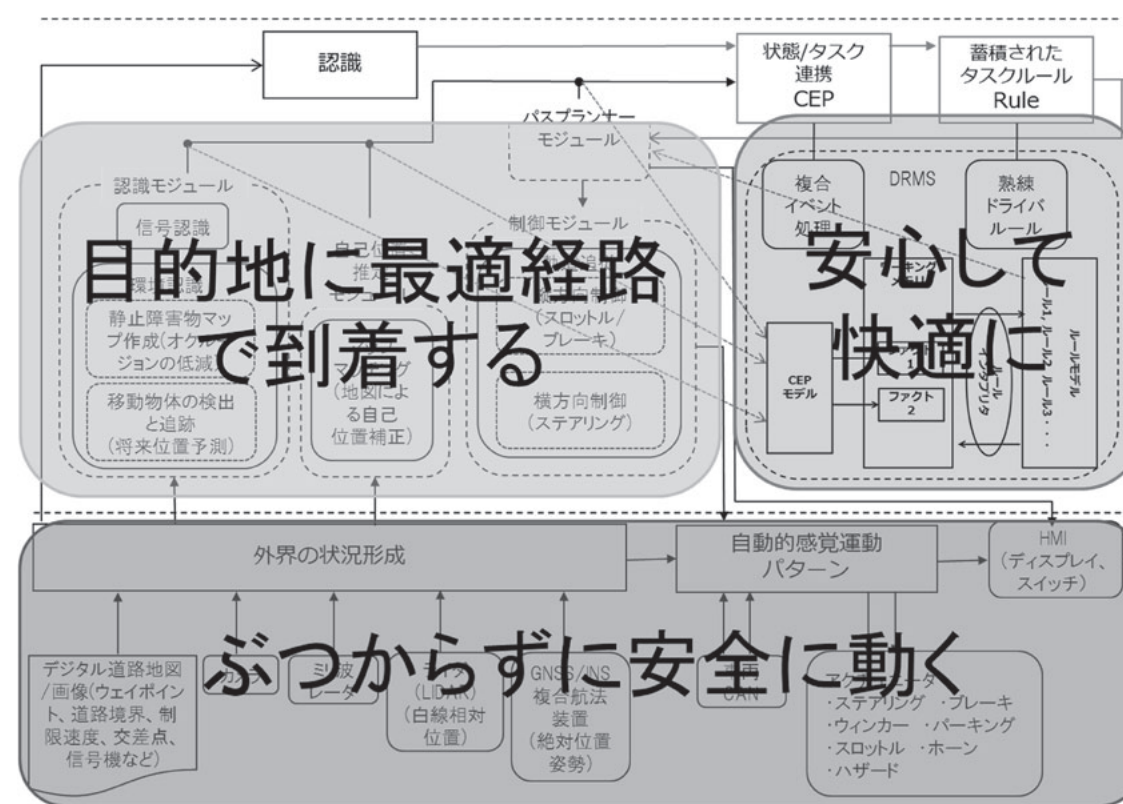
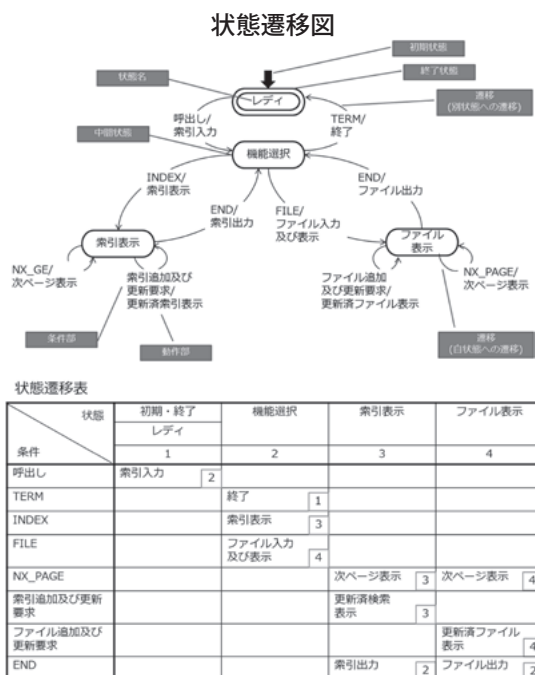


図15 SRモデルとSA

補足

CEP 状態遷移表記法 ISO/IEC 11411 (JIS X0131)

『言語理論という数学の一分野で提案された“有限状態機械”というモデルはソフトウェアの機能や処理をモデル化するに有効である。このため、有限状態機械の表現法である状態遷移図と状態遷移表がソフトウェアの設計や要求定義などの表記法として長い間広く用いられている。』

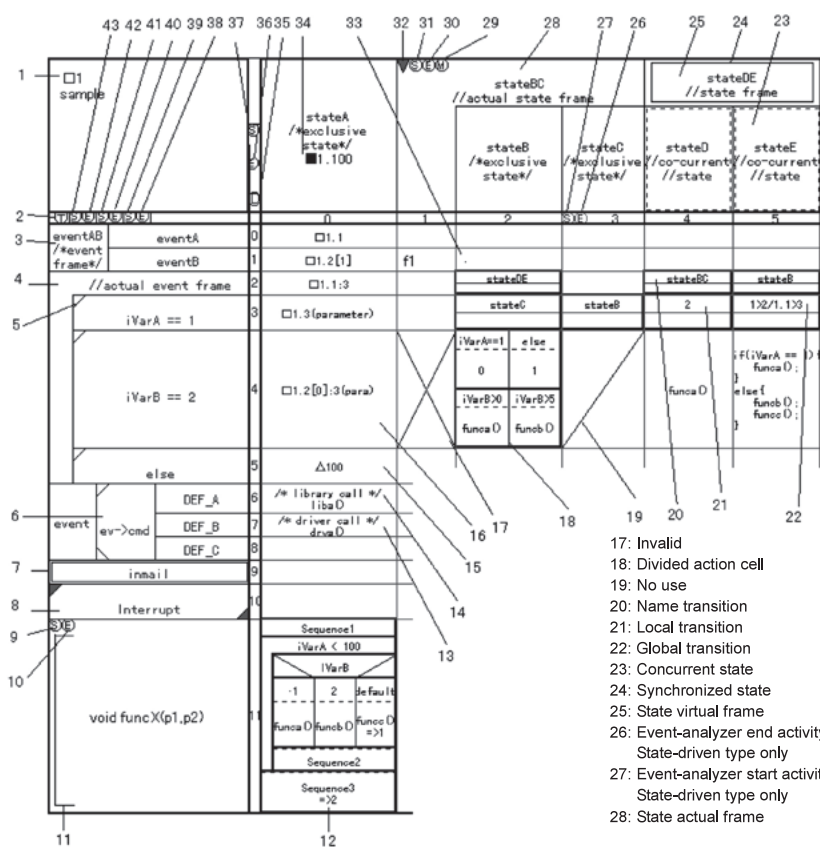


ルール 決定表表記法 ISO5086 (JIS X0125)

『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合せた表。』

制限指定表 (Table 3) with columns for conditions and actions, and rows for specific actions like 'レコードがあるか' and 'レコードを印刷する'.

拡張階層化状態遷移表の書式



- 1: STM definition
2: Trigger activity
3: Event virtual frame
4: Event actual frame
5: If condition
6: Switch condition
7: In-mail
8: Interrupt event
9: Event-hit start activity
10: Event-hit end activity
11: Function-call event
12: Nassi-Shneiderman chart
13: Driver STM call
14: Library STM call
15: Subroutine STM call
16: Event-hierarchy STM call
29: State mode activity
30: State end activity
31: State start activity
32: Default state
33: Don't care
34: State hierarchy STM call
35: Dispatch activity
36: STM main end activity
37: STM main start activity
38: Flag analyzer end activity
39: Flag analyzer start activity
40: In-mail analyzer end activity
41: In-mail analyzer start activity
42: Event-analyzer end activity
43: Event-analyzer start activity

拡張指定表

表7 控除分析 (Table 7) with columns for exclusion codes and rows for exclusion levels and actions.

拡張指定表から制限指定表への変換

表7 控除分析 (Table 7) showing the conversion of exclusion codes into a decision table format.

混合指定表

一次更新表13 (Table 13) showing a mixed specification table with columns for actions and rows for various update scenarios.

参考文献

- [1] 竹垣盛一・石岡卓也：知的制御システム：海文堂出版：1990
- [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] 加藤真平：自動運転システムにおける情報処理技術の最新動向：情報処理 Vol.57 No.5 May 2016
- [4] SIP-adus メディアミーティング 平成 27 年 3 月 9 日 内閣府 プログラムディレクター (PD) 渡邊浩之
- [5] <https://ja.wikipedia.org/wiki/Rete> アルゴリズム
- [6] <http://www.ownway.info/Home/cpp/ps/small/1.html>
- [7] 太原育夫：人工知能の基礎知識：近代科学社：2014 年 9 月 30 日
- [8] コレ 1 枚で分かる「人工知能の 3 つのアプローチ」：ITmedia エンタープライズ：2016 年 4 月 11 日
<http://www.itmedia.co.jp/enterprise/articles/1604/11/news037.html>
- [9] 情報システム用語辞典：BRMS：ITmedia エンタープライズ：2008 年 3 月 18 日
<http://www.itmedia.co.jp/im/articles/0803/18/news146.html>
- [10] ビッグデータ時代のキーテクノロジー、「CEP」とは何か：ITmedia エンタープライズ：2012 年 8 月 22 日
<http://www.itmedia.co.jp/im/articles/1208/22/news133.html>
- [11] OPS5 <http://www.ownway.info/Home/cpp/ps/small/1.html>
CLIPS <http://clipsrules.sourceforge.net/>
Drools <http://www.drools.org/>
- [12] 知識表現の比較
<http://www.rsch.tuis.ac.jp/~nagai/ATb/AII-15-2006.pdf>
- [13] 菅沼直樹：自動車の自動運転技術の動向と開発事例：電子情報通信学会誌 Vol.98 2015 年 1 月
- [14] 永井正夫：交通ビッグデータを基にした先読み運転知能の開発：電子情報通信学会誌 Vol.99 2016 年 6 月
- [15] 東京農工大学：「ヒヤリハットデータベース」：
<http://www.tuat.ac.jp/news/20150806154724/index.html>
- [16] JAF：このとき、あなたは何に注意しますか：
<http://www.jaf.or.jp/eco-safety/safety/danger/>
- [17] JARI：予防安全
<http://www.jari.or.jp/tabid/108/Default.aspx>
- [18] 日本経済新聞 2016/5/5 明治大学法科大学 中山幸二教授
- [19] 山崎良兵、高野敦、野々村洸：自動運転車の“頭脳”でデファクトを目指す：日経ものづくり 2016 年 2 月号
- [20] 野辺継男：自動運転の開発動向：情報処理 Vol.57 No.5 May 2016
- [21] 圧勝「囲碁 A I」が露呈した人工知能の弱点：日経新聞 2016 年 4 月 16 日
- [22] 日経エレクトロニクス 2016.06 人工知能、超人間へのロードマップ p31
- [23] 西川良一：デンソー 情報通信技術 1 部 第 4 技術室 室長：フルグラフィックスメータの開発要件：MATERIAL STAGE Vol.16, No.2 2016
- [24] トヨタ自動車 井上秀雄「高齢社会に向けたクルマの知能化」日本バーチャルリアリティ学会誌第 19 巻 3 号 2014 年 9 月 p17-p20
- [25] 戦略的イノベーション創造プログラム (SIP) 自動走行システム研究開発計画 平成 28 年 6 月 23 日 内閣府 政策統括官 (科学技術・イノベーション担当)
- [26] 機械学習とは：過去の経験から未知の事象を判断する仕組み
<http://www.graffe.jp/blog/906/>
- [27] 今さら聞けないディープラーニングの基本、機械学習とは何が違うのか

- <http://www.sbbbit.jp/article/cont1/32033>
<https://ja.wikipedia.org/wiki/ディープラーニング>
- [28] 武井宏将：初めてのディープラーニング：リックテレコム：2016
- [29] ビジネスルール管理システム (BRMS) 青山学院大学 社会情報学部 森田武史 2013 年 6 月 11 日
[http://www.yamaguti.comp.ae.keio.ac.jp/japanese/2013WI/BRMS%20\(1\).pdf](http://www.yamaguti.comp.ae.keio.ac.jp/japanese/2013WI/BRMS%20(1).pdf)
- [30] [BRMS ソフト] 多岐にわたる業務ルールを管理、利便性向上や BPM 連携急ぐ 岡部 一詩 = 日経コンピュータ 013/06/26
<http://itpro.nikkeibp.co.jp/article/Active/20130610/483561/>
- [31] <https://ja.wikipedia.org/wiki/サブサンプレション・アーキテクチャ>
- 参考： トヨタ自動車 DSSS：安全運転支援システム Driving Safety Support Systems
<http://www.toyota.co.jp/jpn/tech/its/infrastructure/>
- 参考： マツダ 先進安全技術
<http://www.mazda.com/ja/csr/safety/cars/iactivesense/>
- 参考： デンソー 高度運転支援システム
<http://www.globaldenso.com/ja/tokyomotorshow/2015/safety/advanced-driving-support-system/>
- 参考： 高齢者の自立を支援し安全安心社会を実現する自律運転知能システム
<http://web.tuat.ac.jp/~s-innova/research.html>
- 参考： ソフトウェアの状態遷移の構成及びその表記方法 ISO/IEC 11411 (JIS X0131)
- 参考： Specification of single-hit decision table (決定表) ISO5086 (JIS X0125)
- 参考： 渡辺政彦：拡張階層化状態遷移表設計手法：東銀座出版：1998