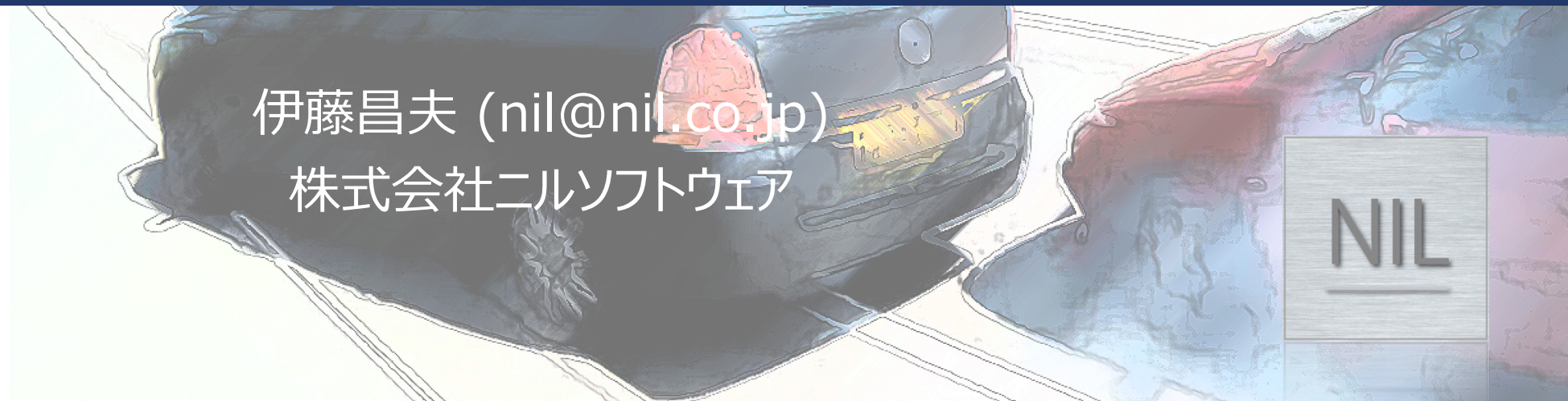
自律走行車と開発プロセス

- AI 組込の観点から -



伊藤昌夫 (nil@nil.co.jp)
株式会社ニルソフトウェア

目次

- ソフトウェアにおける形式検証
- 仕様の一部としてのODD
- AI利用時のプロセスへの影響
  - AIの利用
  - プロセス設計
  - TMカード
- まとめ

ソフトウェアにおける形式検証

Trustworthiness
総合的信頼性（特に安全性）のために

最弱事前条件

- プログラム

$$\{C\}P\{S\} \qquad (1)$$

Cの状態からPを実行するとSの状態になる

- 最弱事前条件（weakest precondition）

$$C \Rightarrow wp(P,S) \qquad (2)$$

最弱事前条件とは，PによってSが成立する上で，最も制約の弱い（少ない）条件

これまでの取り組み：形式検証

形式検証とは，「数学的なモデルを用いて，作られたプログラム（実装）が仕様と合致していることを確認する」こと

【補足】
多くの安全系の規格は，形式手法を使用することを推奨しています．
ISO 26262（自動車），EN 50128（鉄道），DO-178C（航空機）

形式検証の主たるアプローチ：

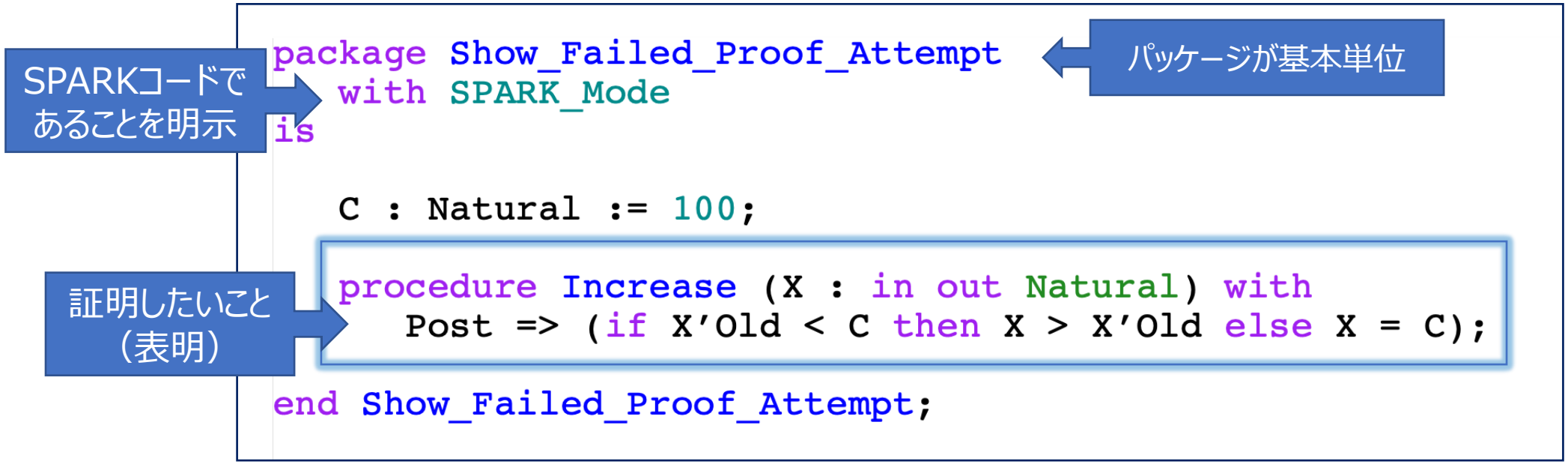
LTL：線形時間論理

- モデル検査	取り得る状態の網羅．望まない状態にならないこと（Safety），望む状態にいつかなること（Liveness) を検出．規模の問題がある．プロパティはLTL式(*)で記述．
- 演繹的アプローチ	プログラムの正しさを証明できる．但し，証明可能な範囲には制限がある（証明器の発展により制限は緩和される）．
- 抽象解釈	静的コード解析．検証プロパティの指定は基本的に不要．偽陽性を検出する場合がある．

Ada … SPARK

- Ada
  - 強い型付けを特徴とする静的なプログラム言語（最初の規格化は1983年）
  - 仕様と実装を分離
  - 言語設計時点で，開発環境も考慮（e.g. APSE）
  - 適用例：Boeing 777
- SPARK
  - Adaの部分的な拡張：「契約」的設計（AdaのAspects利用）
  - フロー分析／実行時検査／正しさの証明（一階の述語論理記述による）
  - 補助手段として，他に契約ケース・ゴーストコード等がある
  - 適用例：Euro Fighter，C130J

SPARK形式検証例：仕様部



仕様部：手続き Increase と事後条件

事後条件では，手続きが終了したときに成立しているべき条件を示す．ここでは，入力*X*の値が変数*C*より小さいならば，出力*X*は，入力*X*より必ず大きくなり，それ以外の条件では変数*C*の値に等しいことを仕様として示している．これが，手続き利用者との「契約」となる．
なお，*X'Old* は，この手続きを実行する前の自然数*X*の値のこと．*X* は *in out* パラメータであり，値が変化する

違いは何か？

- そもそもが前提条件が難しい
  - 自律走行車においては，ODDの設定は典型的な例である

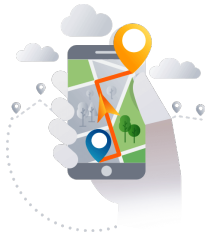
$$C \Rightarrow wp(P, S) \qquad (2)$$

仕様の一部としてのODD

Definition of ODD in J3016:2021 and SHs

Definition

3.21 Operating conditions under which a given *driving automation system* or *feature* thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics.



Geography

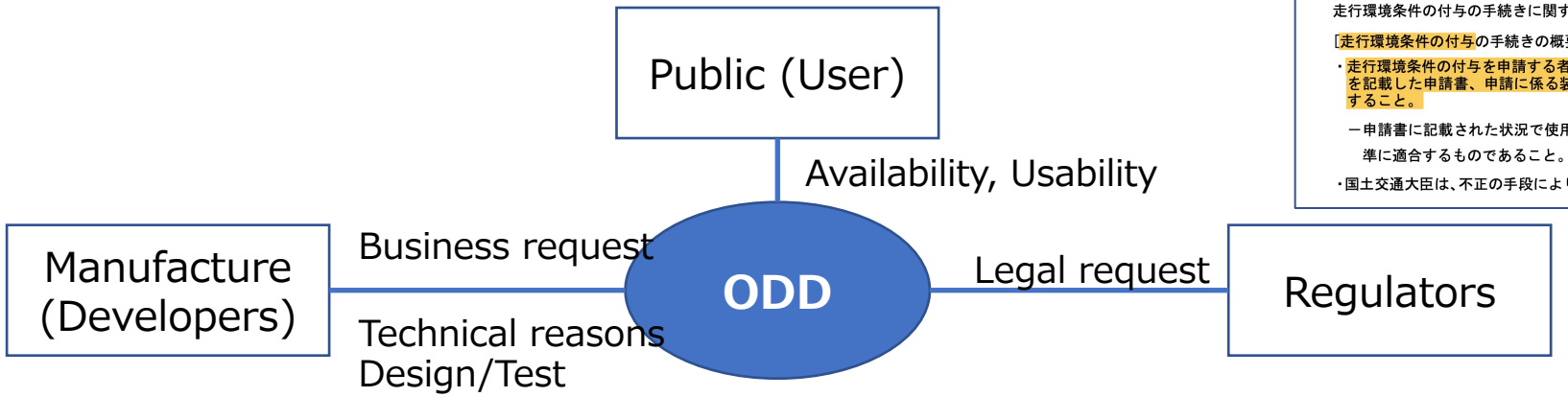
Environment

Time-of-day

Car, Pedestrian ..

ODDのさまざまな利用者

開発側に関わらず，利用者や法的執行機関もまた理解する必要がある



（２）道路運送車両法施行規則の一部改正

走行環境条件の付与の手続きに関する規定を追加するほか、所要の改正を行う。

［走行環境条件の付与の手続きの概要］

・走行環境条件の付与を申請する者は、「自動運行装置が使用される状況（場所、気象、交通等）」等を記載した申請書、申請に係る装置が以下に適合することを証する書類等を国土交通大臣に提出すること。

－申請書に記載された状況で使用されるものと仮定した場合おいて、自動運行装置に係る保安基準に適合するものであること。

・国土交通大臣は、不正の手段により付与を受けたとき等には、条件の付与を取り消すことができる。

ODD記述法に関する幾つかの規格・文書

#	ID	Title	Issued by	Date
1	BSI PAS 1883	Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specification	BSI	31.Aug. 2020
2	WISE	Operational Design Domain for Automated Driving Systems Taxonomy of Basic Terms (Operational World Model Ontology - Part 1 Road & Part 2: Road Users, Animals, Other Obstacles, and Environmental Conditions)	Waterloo Intelligent Systems Engineering (WISE) Lab	Jul.2018
3	AVSC	AVSC Best Practice for Describing an Operational Design Domain: Conceptual Framework and Lexicon	Automated Vehicle Safety Consortium	Apr.2020
4	OpenODD	(Ontologies and ODDs at Five)	(Lain Whiteside)	(Apr. 2020)

Example (from AVSC)

The system is designed to operate on the road network in the urban center of Detroit, Michigan on all streets with a speed limit of 35mph or less. Its boundary is constrained by I-75 to the north; I-375 to the east; M-10 to the west; and Atwater Street along the Detroit River to the south. The areas around the Detroit Police Department and Department of Public Safety are excluded from this ODD. The system is capable of operating during daylight hours when the sun is at or above the horizon. It can operate in fair weather, including wind gusts up to 31 mph, light rain, and light snow, provided the road surface is not covered by snow nor standing water. It recognizes and understands all signage and traffic control devices inside this ODD. Work zones are coordinated with the local transportation department and excluded from the route network as needed. And the System needs cellular connection.

Geofence Region

WISE：ODD要素

- ADS for Level 4 and Level 5

Top Level	2nd Level
Road structure	road type and capacity
	road surface type and quality
	road geometry
	cross-section design
	traffic control devices
	pedestrian crossing facilities
	cycling facilities
	junctions
	railroad level crossings
	bridges
	tunnels
	driveways
	temporary road structure

Top Level	2nd Level
Road users	ground vehicles, including their occupants
	animal riders
	pedestrians
	traffic control persons
Animals	small
	medium-size
	large
Other obstacles	-
Environmental conditions	atmospheric conditions
	lighting conditions
	weather-related road surface cond.

WISE: 例を割り当ててみる

Top Level	2nd Level	values
Road structure	road type and capacity	Local roads / NC
	road surface type and quality	NC
	road geometry	NC
	cross-section design	NC
	traffic Control devices	NC
	pedestrian crossing facilities	NC
	cycling facilities	NC
	junctions	NC
	railroad level crossings	NC
	bridges	NC
	tunnels	-
	driveways	NC
	temporary road structure	NO

Top Level	2nd Level	values
Road users	ground vehicles, including their occupants	NC
	animal riders	NC
	pedestrians	NC
	traffic control persons	NC
Animals	small	?
	medium-size	?
	large	?
Other obstacles	-	NC
Environ-mental conditions	*atmospheric conditions*	fair, light rain, light snow, no gust
	lighting conditions	daytime
	weather-related road surface cond.	not covered by snow

検証可能性

- 定義： 真或いは正しいことを証明できるという事実（*the fact of being able to be proved to be true or correct ; Cambridge dictionary)*
  - 選挙に於いては，自分の投票が変更されたかどうかを確認出来れば，*end-to-end* の検証ができる

OEDR: Object and Event Detection and Response

- 検証可能性: Imp (an ADS) ⊧ Spec (an ODD + f_{OEDR} +α)
  - 仕様を記述することができるだろうか．できるとすれば，如何なる記述法で？
- 検証可能性を高める方法のひとつは，合理性の提示である.
  - ひとつの例：Toulminモデルを用いる

UL4600

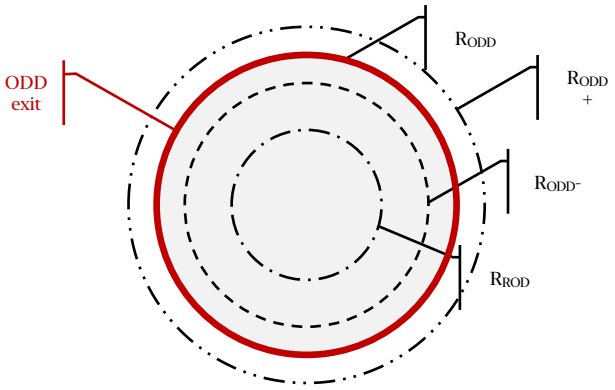
- タイトル: 自律製品を評価するための安全性に対する標準（Standard for Safety for the Evaluation of Autonomous Products）
- 発行： 2020年4月
- 特徴：「論証（argumentation）」に基づくセーフティケースベース
- ODD Clauses (8.2);

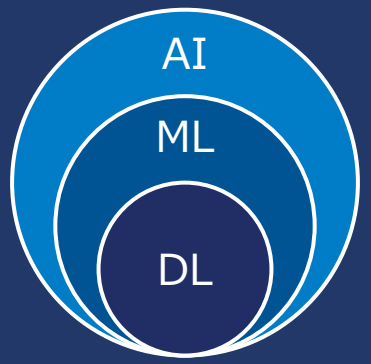
(?) 8.2.1 運用設計ドメイン(ODD)は，受容可能な完全な方法で記述されていること

Partially Yes 8.2.2 ODDは，自律型アイテムが動作する関連する環境面をカバーしていること

NO 8.2.3 ODDからの逸脱は，受容可能な安全な方法で処理されること

NO 8.2.4 ODDの変更を検出し，解決までトレースすること





AIとプロセス

- AI（人工知能）
  - 分類器（除く，シンボルタイプ）
- ML（機械学習）
  - データを用いて，統計的に学習を行う
  - 教師あり／なし，強化学習がある
- DL（深層学習）
  - 多数の層とノードからなり（ニューラルネットワーク），多数のデータを用いて，学習を行う

AIモジュールの一般的課題

開発	生産	組織
• 実験管理 • 限られた透明性 • トラブルシューティング • リソース制約 • テスト	• 依存関係の管理 • 監視とログ記録 • 意図しないフィードバックループ • グルーコードとサポートシステム	• 労力の見積 • プライバシー • データの安全性 • 文化の違い

Arpteg, Anders, et al. "Software engineering challenges of deep learning." *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2018

プロセスを考える上での考慮点

A) ライフサイクル上の位置と特性
- ライフサイクルのどの位置で使用するか

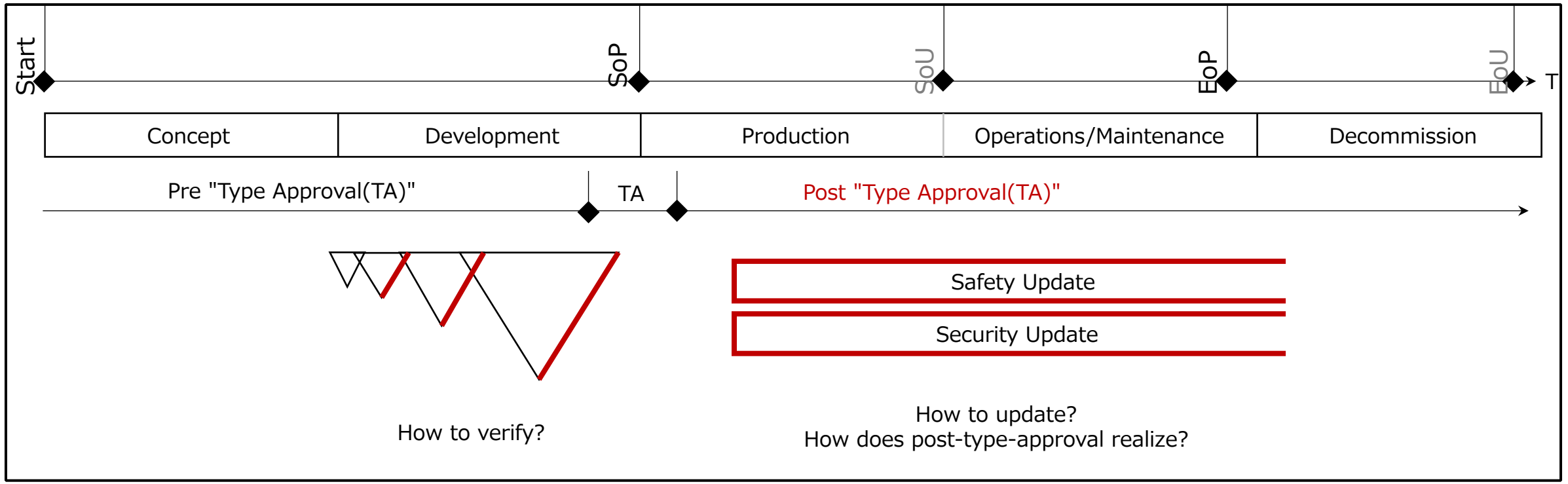
B) AIシステムのタイプ
- 知識／推論／認知

C) 車両における対象アイテムとAIモジュール
- どのアイテムで使用しするか．その機能は.

D) 注意すべき（特有の）品質特性
- 一般的なISO25000(SQuaRE)の前に

A）ライフサイクル上の位置と特性

［自律走行車］前提となるODD記述
［AI利用］　データ管理プロセス／モデル管理プロセス
［両者］　型式認証後の更新，検証



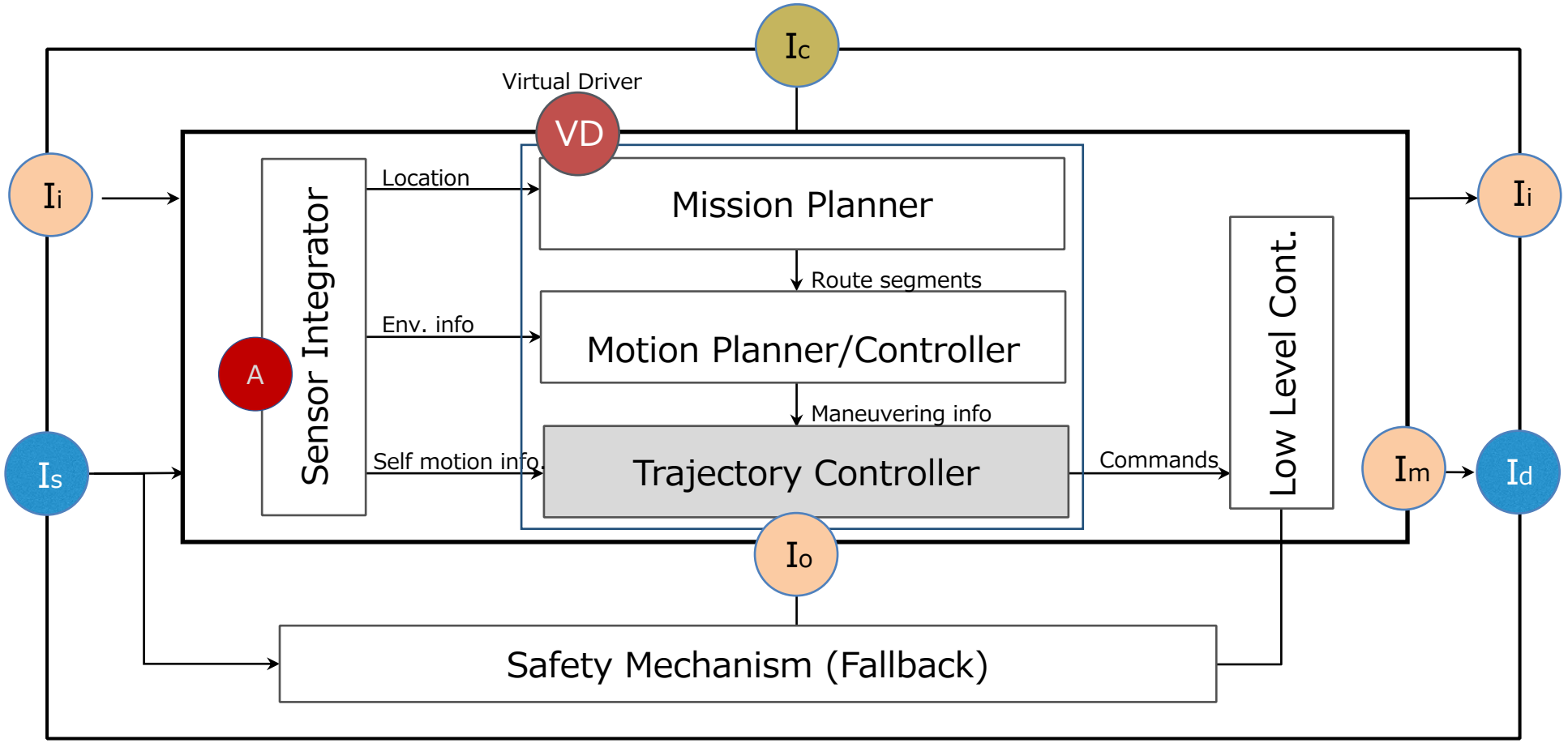
劣化の考え方が異なる：機械は経年，即ちそのモノの劣化，AIでは，環境変化に伴う不適合，モノは変化しない

B)どのようなAIシステムか

出典：Forbes, "AI Knowledge Map: How To Classify AI Technologies, 一部のみ
https://www.forbes.com/sites/cognitiveworld/2018/08/22/ai-knowledge-map-how-to-classify-ai-technologies/?sh=7b35c5017773

C) どこで使用するか：構造とAIモジュールの位置

ADS-DV(SAE L5)



D）注意すべき品質特性
包括的（DIN92001），倫理上（EGTA）

- 三本柱（DIN92001）
  - 機能／非機能，ロバスト性（-2，敵対／劣化），理解可能性
- EGTA

EGTA カテゴリ	構成原則
人間エージェンシーと監視	基本的権利
	人間エージェンシー
	人間による監視
技術的ロバストネスと安全性	攻撃に対するレジリエンスとセキュリティ
	フォールバックプラン
	一般的安全性
	正確さ
	信頼性と再生産性
プライバシーとデータガバナンス	プライバシーとデータ保護
	データ品質とデータの完全性
	個々のデータへのアクセス

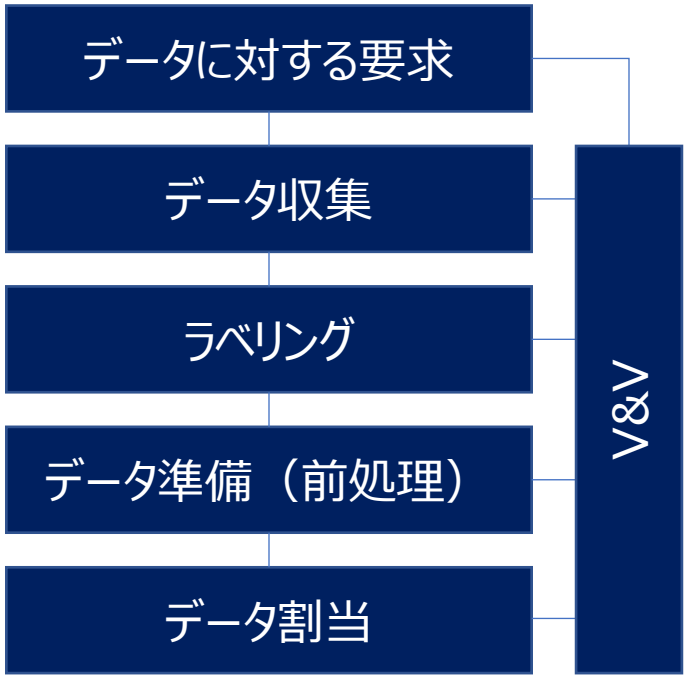
EGTA カテゴリ	構成原則
透明性	追跡可能性
	説明可能性
	コミュニケーション
多様性，被差別と公平性	不公平なバイアスの除去
	アクセス可能性とユニバーサルデザイン
	ステークホルダの参加
社会／環境の良い状態	環境負荷を上げずに継続可能である
	人間エージェンシー
	人間による監視
説明責任	監査可能性
	悪影響の最小化と報告
	トレードオフ
	救済

EGTA:=Ethics and Guidelines on Trustworthy AI（EASA）

AI利用による追加が必要なプロセス例
データ管理プロセス

EASA concept paper First usable guidance for level 1 machine learning applications

- データは，これまで以上に重要な役割を果たす

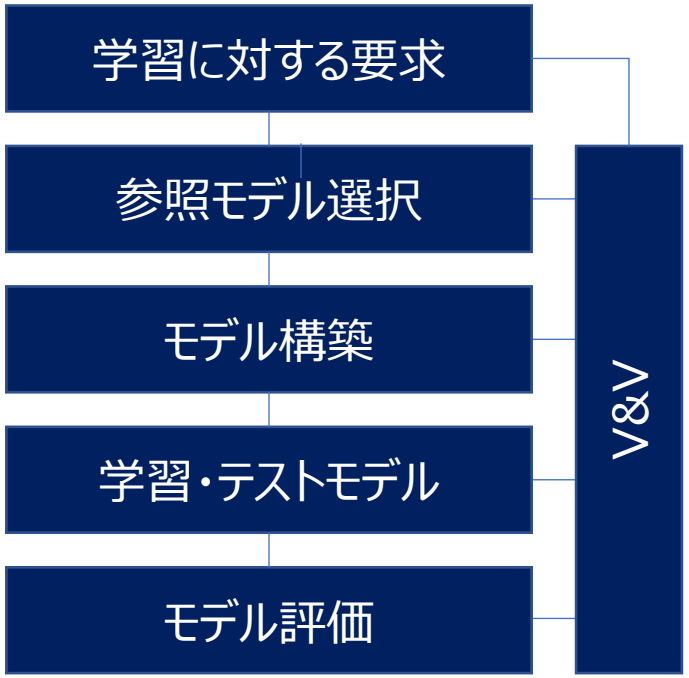


- データの正確性
- データの解像度
- 注釈付きデータの品質
- （通信ネットワークを介して）保存／処理／送信中の破損がないこと
- データの出所を特定する機能（トレーサビリティ）
- データが使用目的の期間に適用可能であるという信頼度（適時性）
- 使用目的（完全性と代表性）をサポートするために必要なデータであること
- データの形式

AI利用による追加が必要なプロセス例
学習管理プロセス

EASA concept paper First usable guidance for level 1 machine learning applications

- 適切なモデルを採用すること



- 機能要件，非機能要件（WCET）
- モデルファミリとモデル選択
- 学習アルゴリズムの選択
- パフォーマンスおよび安全性メトリックへのリンクを説明するコスト/損失関数の選択
- モデルのバイアスと分散のメトリックおよび許容レベル
- トレーニング環境（ハードウェアおよびソフトウェア）
- モデルパラメータの初期化戦略
- ハイパーパラメータとパラメータの識別と設定
- トレーニング，検証，およびテストデータセットで期待されるパフォーマンス

ライフサイクルプロセスの一部
AIモジュール（AIM）開発段階

#	プロセス名
P.1	要求識別
P.2	学習アルゴリズム開発
P.3	学習／評価データ関連
P.4	学習
P.5	AIモジュールの評価

SCSC-153B，UL4600，DIN 92001を利用

アイテム

AIM

Model Selection and Customize, Model Creation
An appropriate algorithm type is used. (S:COM3-1)

AR Adversarial Robustness
CR Corruption Robustness
LIA Likelihood & Impact Analysis

(functional/non-functional)
P1
P2/3
P2
P2/3
P4.1
P4.2
P5

Requirements Specification of AIM
Algorithm and Data (S:COM1-3)
Creating Training Data (S:COM1-1,1-2)
COM1-3
Training
Evaluation of Training
Verification
Transfer for operation
Acceptable Data (UL.8.5.3)
P3
P3.1
P3.2
Data Evaluation (DIN 3.3)
Functional Evaluation (S:COM2-1)
Non-Functional Evaluation (S:COM2-2)
Algorithm performance is measured objectively (S:COM2-3)
Creating Verification Data

Performance boundaries are established and complied with（S:COM2-4）

AR/CR Risk Identification (DIN 5.2.2)
AR/CR Threat Model Analysis, LIA (DIN 5.3.2)
AR/CR Robustness Evaluation（DIN:5.3.4, DIN: 5.4.4）
Robustness Analysis
R: DIN 92001-2
Test Planning（DIN: RAR_EVL.1, RCR_EVL.1）

Horizontal Process
Typical errors are identified and protected against (COM3-2)
The algorithm's behaviour is explainable (COM3-3)
Risk Analysis（DIN RGE）
HW/SW Separation

プロセスに関する簡単な代数

We will consider the above in detail. Let's say design is D and coding is C. The test is T. Abstractly, we can think the following order.

$$D < C < T \quad (1)$$

The symbol '<' indicates the execution order here.
Now, decompose T1 as follows.

$$T := T_1 < T_2 < T_3 \quad (2)$$

Here, T_1 is test case creation, T_2 is test code creation, and T_3 is test execution. Simply, (1) becomes:

$$D < C < (T_1 < T_2 < T_3) \quad (3)$$

However, if you think about test first, you can also:

or,

$$D < T_1 < T_2 < C < T_3 \quad (4)$$

$$T_1 < D < T_2 < C < T_3 \quad (5)$$

Next, consider the process instance. Now assume that $\{di_1, di_2, .., di_n\}$ is an instance of process element D.

$di_1, ..., d_{ik}$: D_{P1} is a critical element, and I want to confirm its feasibility early. On the other hand, $di_l, ..., di_n$: D_{P2} is an easy element so that we can design it later.

If you take a strategy to tackle difficult issues first, we can get the sequence showing below:

$$D_{P1} < C_{P1} < T_{P1} < D_{P2} < C_{P2} < T_{P2} \quad (6)$$

This is a kind of incremental approach. It is easy to find for us by using parenthesis.

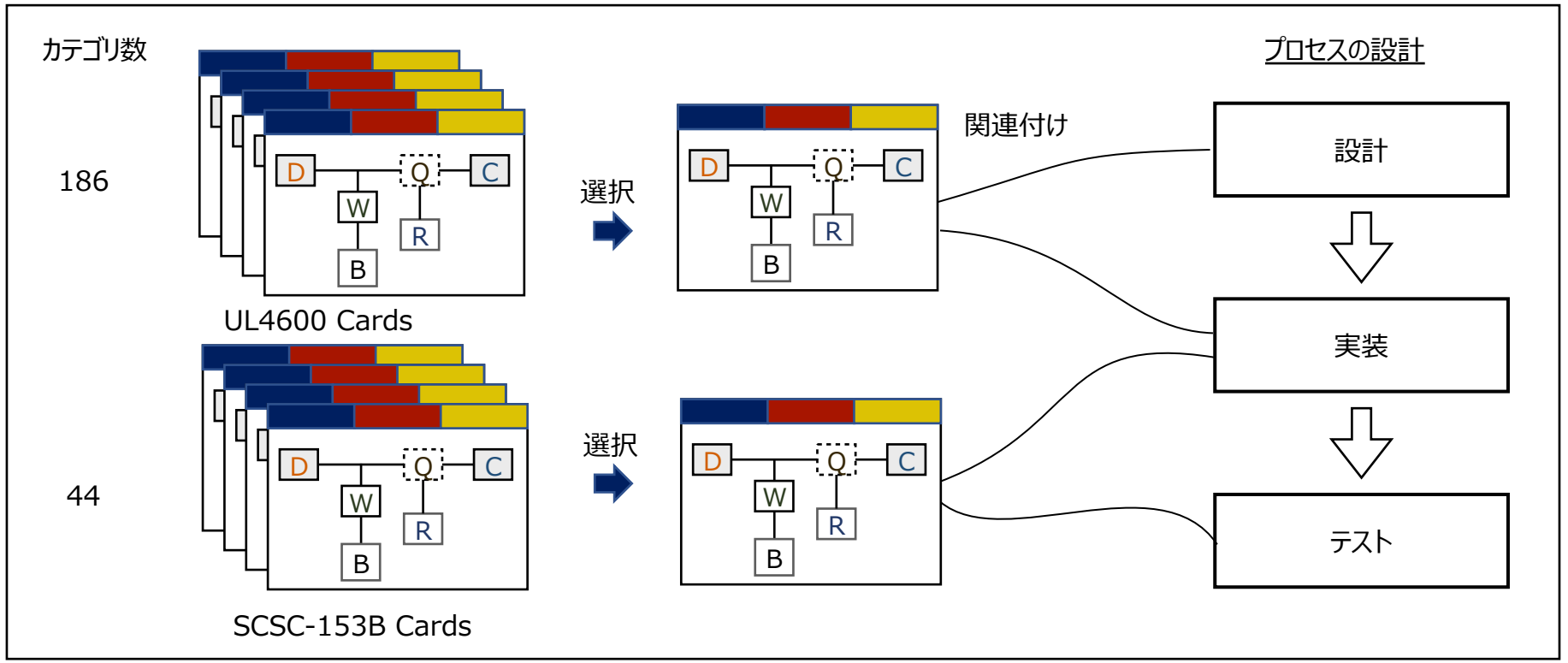
$$(D_{P1} < C_{P1} < T_{P1)}) < (D_{P2} < C_{P2} < C_{P2}) < ... \quad (7)$$

The work products are the same in both cases. The only difference is the choice of the process designer.

That is, the work product definition of each activity does not uniquely determine the order in which the activities are enacted.

TM Card

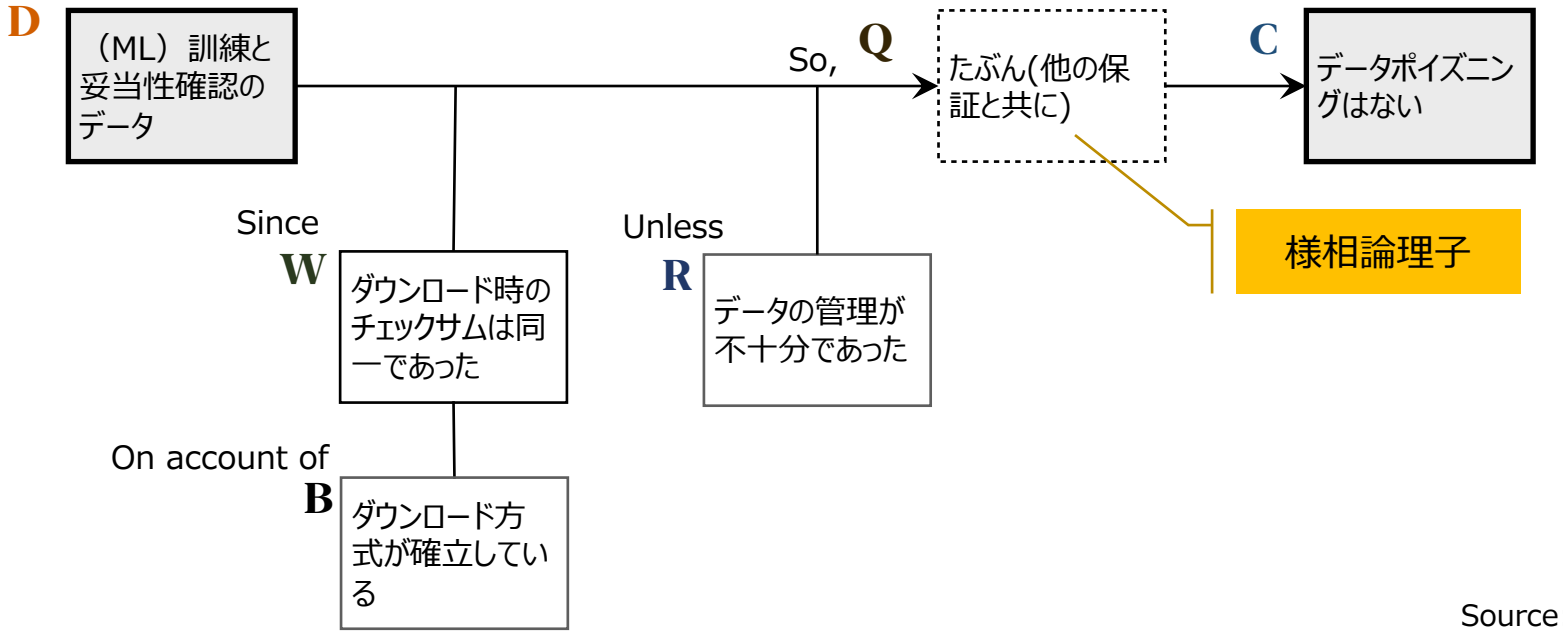
- 注意すべきプロセス要素を，トゥールミンモデルに従い，カード化する
  - 使用するのは，SCSC-153B，UL4600，DIN 92001



SCSC-153A: a sample TM card

AC (Autonomous Car)	ML (Machine Learning)	D, I

COM1-1：データは適切に取得および管理されること
…完全なデータセットを取得する場合「完全なデータセットを外部から取得する場合は、「データポイズニング」の対象にならないように注意する必要があります．たとえば、悪意を持って作成されたわずかな数のサンプルがバックドアを作成する可能性があります」。ここでは、インターネットからダウンロードした情報（チェックサムなど）の信頼性を確認するために使用したのと同じ手法が役立つ場合があります．



Source: SCSC-153B

まとめ

- 自律走行車の開発プロセスについて考えた．特にAIを使用する場合について．
- 自律走行車の前提となるのは，ODDであるが，必ずしも定まった記述法が存在するわけではない．安全性確保に強く影響する検証を困難にする原因のひとつである．
- 自律走行車もAIは，新しい技術であり，現在多くの規格を含む文書が発行され，或いは準備されている．
- さまざまな条件下で，（如何なる場合に於いても）開発プロセスを一意に定めることはできない．解決のためのアイデアとして，ここではTMカードについて説明した．

まとめ（将来のために）

- 先に示した式のように，C（ODD）を導出した方が，正確な記述が可能となると考える．このとき，S は安全な状態であり，Pは，それを逆にCに変換する
- 今の技術できることを数えると信号や標識の認識となる．しかし，重要なのは，現時点での法的制約を知ることである
  - 視覚優位なのは，歴史的な積み重ねに過ぎない
  - より確実に知る方法はある（車に対しても人に対しても）
- 現時点では，C も P もある種の尤度を持っている． 従って安全性を議論しても，ALARP を満足することができない

ありがとうございました