NIL

機能安全要求の導出方法

– 初期アーキテクチャと時間間隔を中心に –

株式会社ニルソフトウェア
伊藤 昌夫

概要

- 乗用車の搭載システムの概念段階での安全性を考える．ISO26262は，最後に機能安全仕様を作成することを要請している．中心的な課題は，機能的冗長性を実現した初期アーキテクチャと，各種時間間隔である
- （もちろん）規格中で明確な解法は示されていない．本発表では，具体的な手法として，我々が用いているアプローチを説明する．
- 最初に機能的冗長性を与える機能安全機構を持つ初期アーキテクチャの抽象モデルを示す．次に，安全ゴールに応じた時間記述を行うためにAADLを用いる．ハザードとエラーは密接に関係しており，AADLのエラーモデルは，適切な初期アーキテクチャの記述においても有用である．

導入

NIL

- ISO26262における基本的な概念フェーズの流れ

概念フェーズのステップ

アイテム定義（3-5）	アイテムを定義し，環境との相互作用を記述する
安全ライフサイクルの開始（3-6）	新規のアイテムと既存のアイテムの改修を区別した上で，安全ライフサイクルを開始する
ハザード分析とリスク評定（3-7）	アイテムのハザードを識別し，カテゴリ分けをする．安全ゴールを設定する
機能安全概念(3-8)	機能安全要求を設定する．アイテムの初期アーキテクチャにおいて，その実現を示す．

前々回

前回

今回

（注）括弧内は，定義している場所を示す．例えば，3-5は，3分冊の5章を示している

導入：対象

NIL

- 機能安全要求

要すれば，以下を記述する（3-8.4.2.3）

■ 動作モード	対象としている動作モード	
■ 故障耐性時間間隔	故障発生から危険状態までの時間間隔	(B)
■ 安全状態	対象が達すべき安全状態	
■ 緊急動作時間間隔	故障発生から，安全状態移行までの時間間隔	(B)
■ 機能的冗長性	対象が持つべき機能的冗長性	(A)

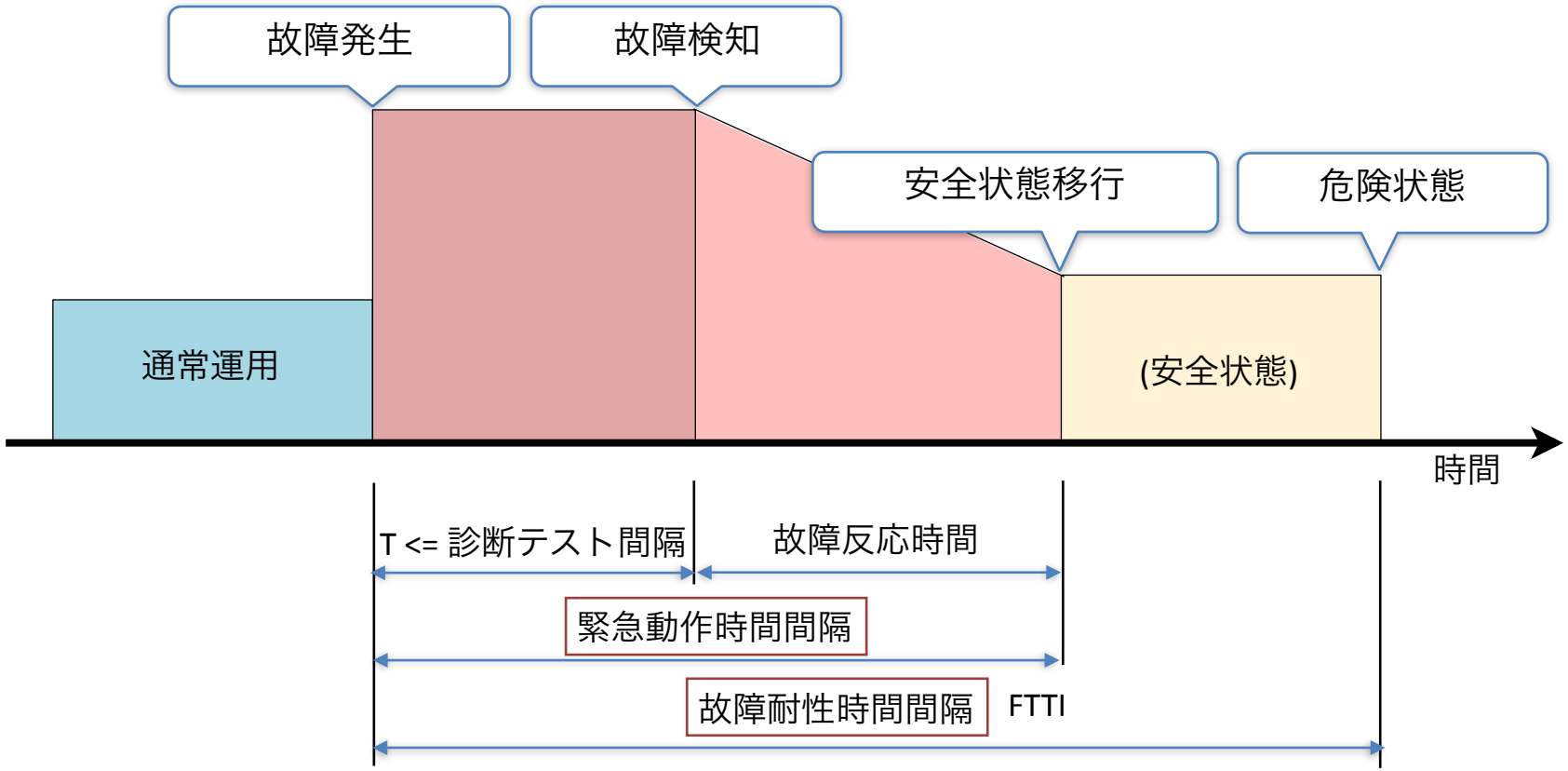
<u>今回の発表のポイント</u>

（A）抽象的な機能安全機構

（B）AADL記述によるフロー解析，エラー記述

導入：対象

- 故障モデル



Fault reaction time and fault tolerant time interval (ISO26262-1 Fig.4)を改変

機能的冗長性

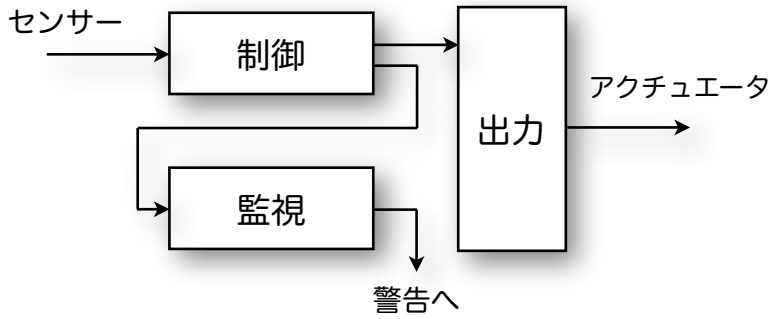
NIL

- 機能的冗長性は，対象となるアイテムの要素に，機能安全機構を対置することで実現する
  - 多くの場合，ASIL分割を行い，機能安全機構は対象に対して，小さくないASILの値を与える（今回は，ASIL分割については触れない）

機能安全機構の例

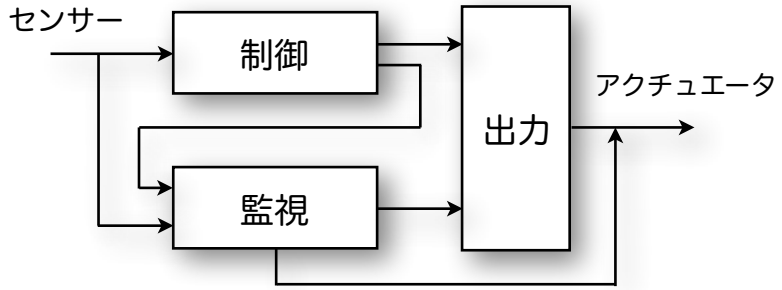
NIL

(A)



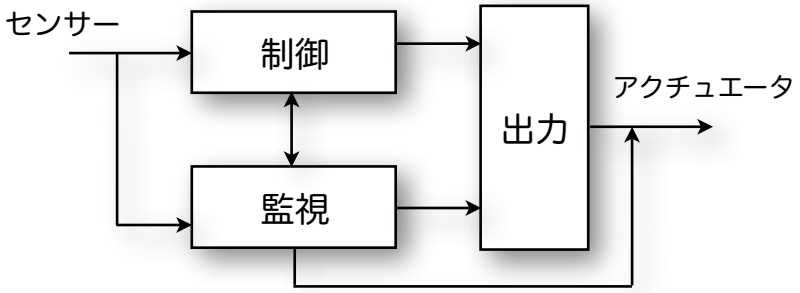
制御出力の監視のみを行う

(B)



センサーからの入力及び制御出力を監視し，要すれば出力をオーバライドする

(C)



センサーからの入力及び制御出力を監視し，要すれば出力をオーバライドする

凡例

制御　対象となる制御

監視　対象となる制御部分を監視する機能安全機構

抽象的な機能安全機構

NIL

- 最初に対象を基本形として記述する

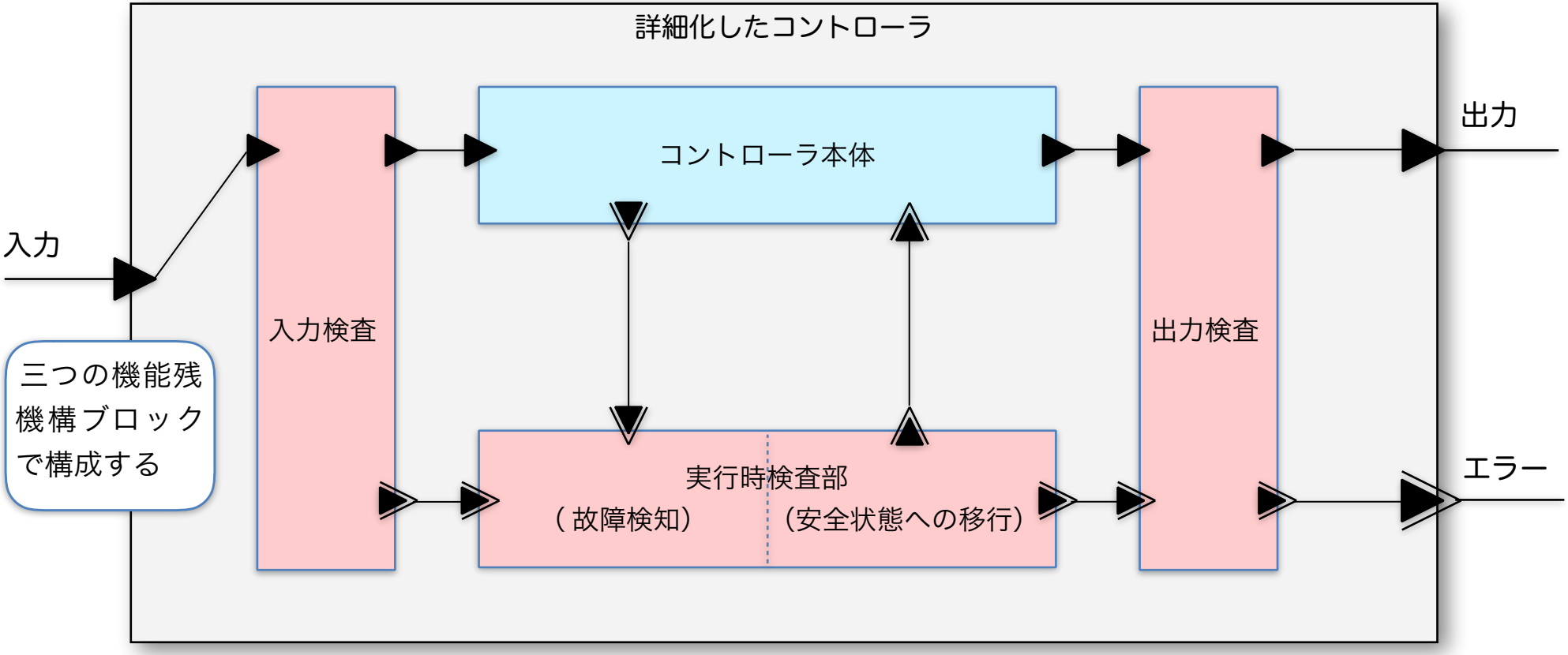
本論の範囲外であるがアイテムスケッチから導ける

【注記】以降では，議論を先取りして，AADLの図式を一部用いる

抽象的な機能安全機構

- 基本形を詳細化する．ここでは概念段階で考えるべき必要な要素を含んでおり(*)，対象（具体的にはASIL）に従って簡略化できる

（＊）もちろん，３つの機能安全機構は，レイテントフォールトを持つ可能性がある．しかし，このパーツも同様の詳細化を加えることができる．無限後退するが，どこで停止するかは後の設計段階で定める



初期アーキテクチャの記述

NIL

【参考】AADL（Architecture Analysis & Design Language）

- SAE航空宇宙部門の規格で，名前の通りアーキテクチャの分析・設計用の言語である．2004年に初版が発行され，最新はAS5506 Rev. Bである
- 基本的には，テキストで記述する．対応する図式も定めている（AS5506B Appendix D）．Annex には，ふるまいモデル（Annex D, AS5506/2），エラーモデル（Annex E，AS5506/1A）を含んでいる．
- ベースは，Ada言語であり，Ada言語の特徴である強い型付けを持ち，仕様部と本体部の分離と言った開発時の特徴を維持している．

初期アーキテクチャの記述


```
system implementation comp0.i
        subcomponents
                c : system pcontroller
                        {ISO26262::ASIL => LEVEL_B;};

                i : system pfsminp.i;
                s : system pfsmcre.i;
                o : system pfsmout.i;
        connections
                c0 : port i.p_out -> s.p_in;
                c1 : port s.p_out -> o.p_in;
                ce : port o.p_err -> p_err;
        annex EMV2 {**
                use types errorlibrary;
                use behavior
NILErrorModelLibrary::Basic_behave;
                …
                -- state transition --
                composite error behavior
                states
                        [o.failed]->failed;
                end composite;
        **};
end comp0.i;
```

全体（詳細化したコントローラ）を示す

定義したISO26262用プロパティセットの利用

入力検査部，実行時検査部
出力検査部

全体の入出力

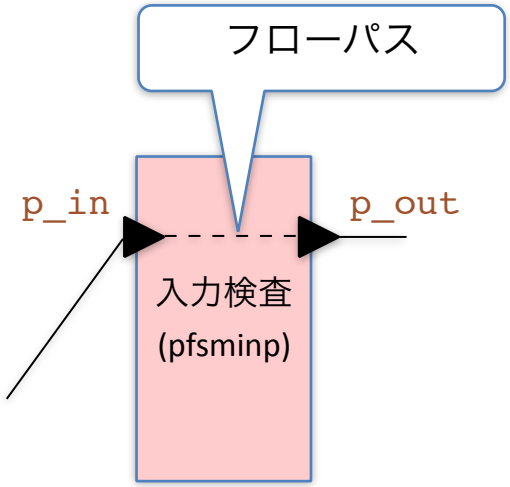
エラーAnnexの利用

今回定義したふるまい

遅延時間の推定

```
system pfsminp
        features
                p_in  : in  event data port;
                p_out : out event data port;
        flows
                fl10 : flow path p_in -> p_out
                { latency => 1 Ms .. 4 Ms; };
```

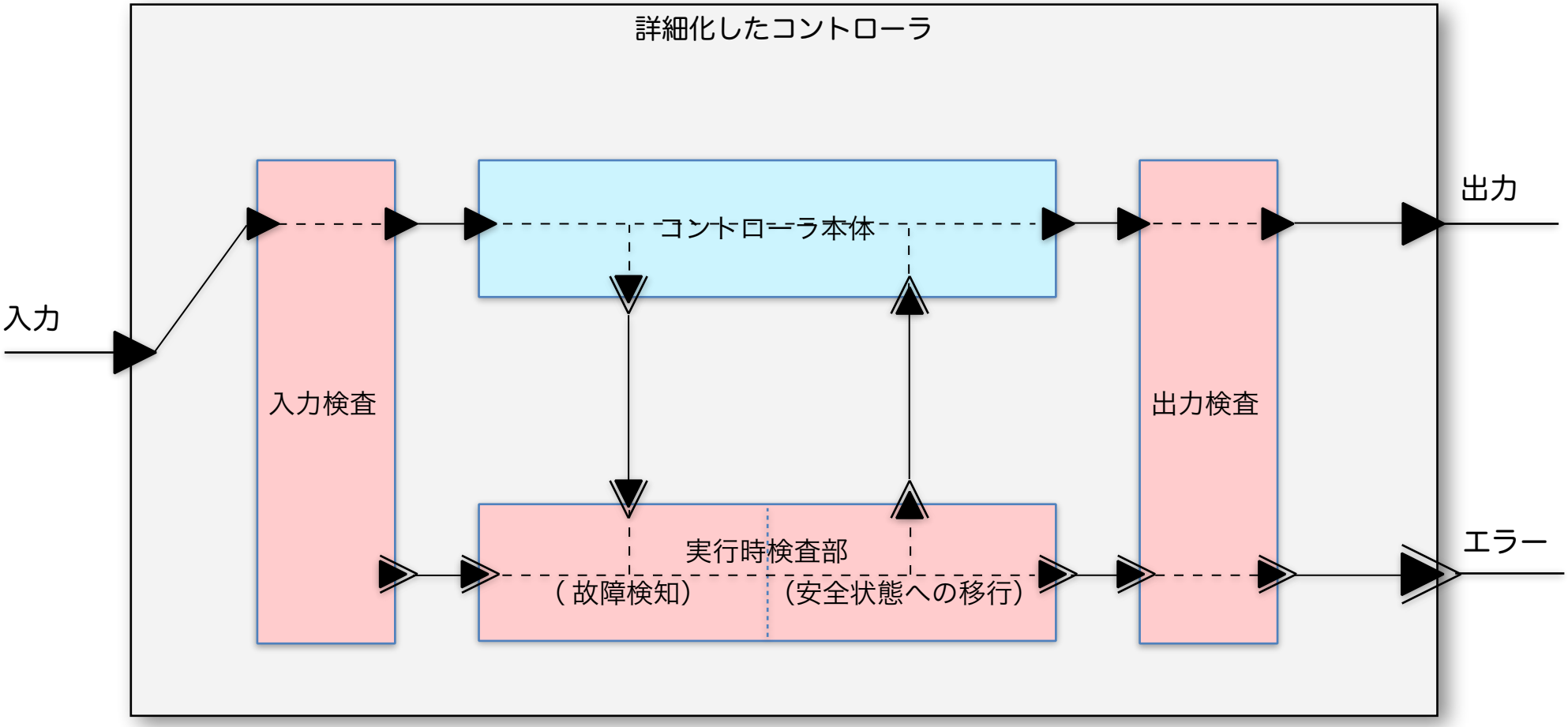
フローパスにおける遅延



フローパスとして，想定した遅延を記述できる

緊急動作時間間隔の計算

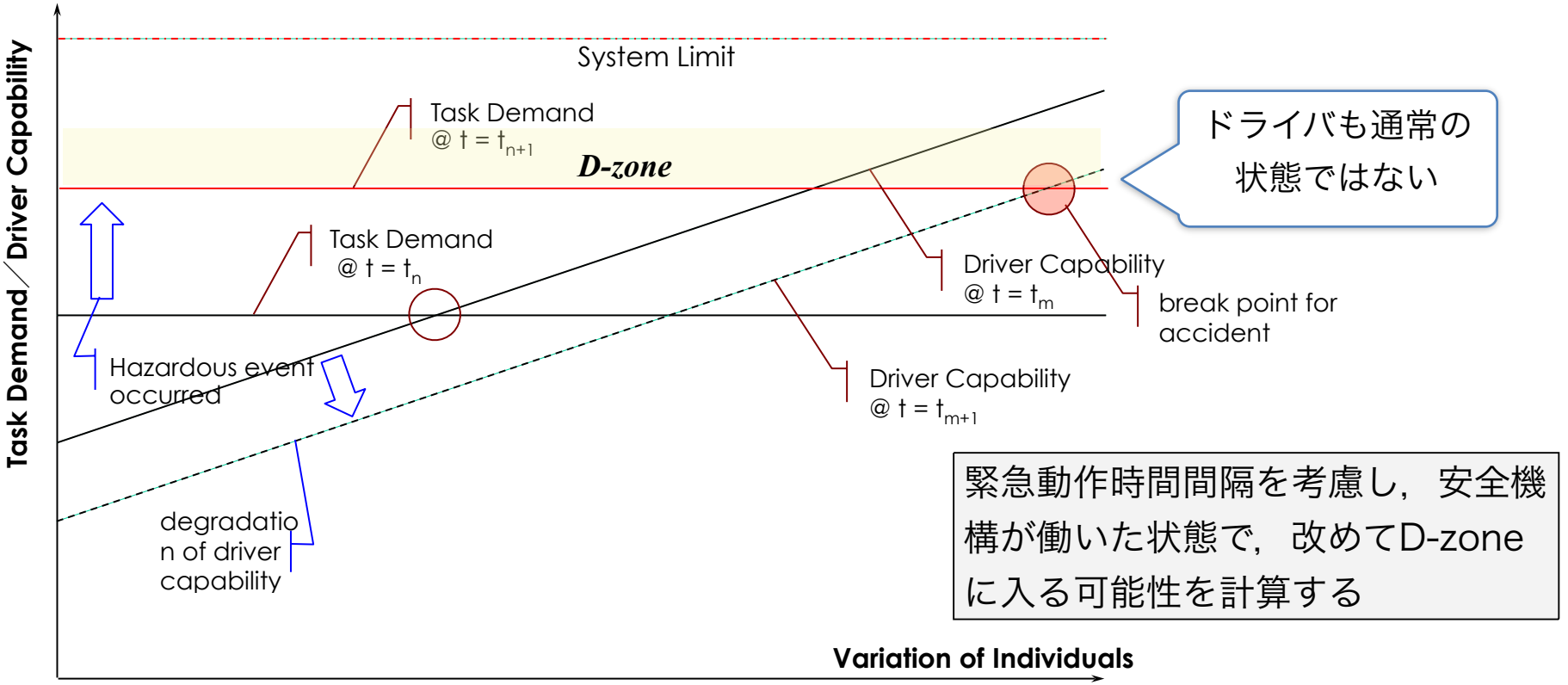
- 緊急動作時間間隔の計算に必要なのは，故障検知・安全状態を経由するフローパス．もちろん，最終的には「アクチュエータ」の動作完了が必要



FTTIの計算

NIL

- 安全状態に移行しても，機能の縮減を伴う場合は，危険状態となる
- 計算には本論外であるが，当該ハザードの生じる状況（ASIL計算のために与えてある）と運転可能性が必要である.



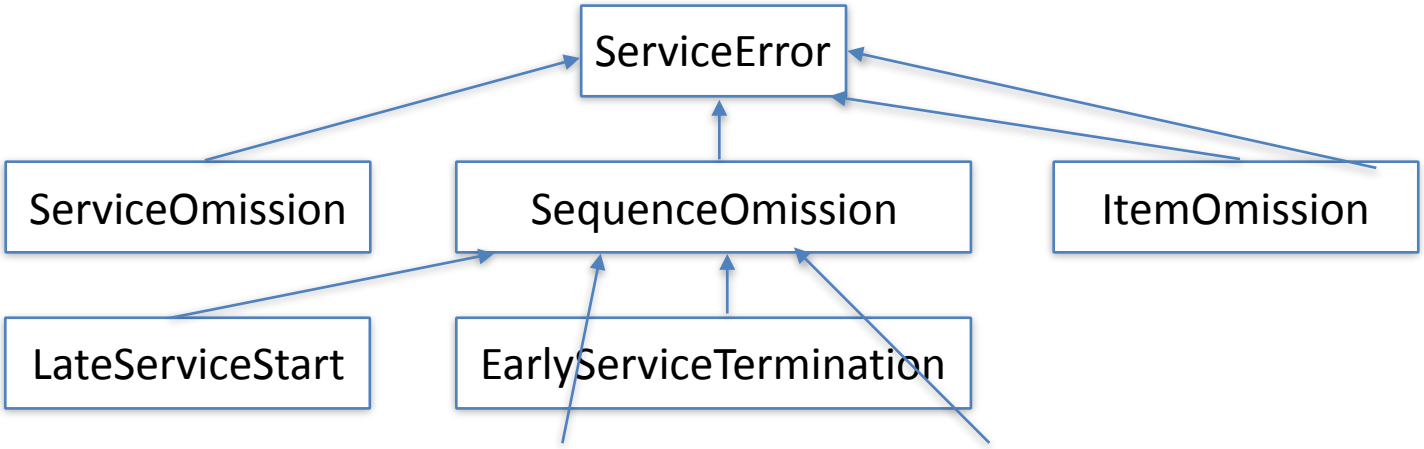
エラー型

NIL

- エラーモデル（EMV2）では，様々なエラーが定義されている

```
CommonErrors: type set { ServiceError, TimingRelatedError,
ValueRelatedError, ReplicationError, ConcurrencyError};
```

- それぞれは，更に拡張され階層を持っている．サービスエラー型の一部を以下に示す



サービス関連のエラー型階層（大きくは，上記のOmissionとCommisionに分かれる）

エラーの扱い

NIL

- 初期アーキテクチャに応じたエラー型を選択する
  - 例えば，AADLの持つレプリケーション関連エラー型は，新規アイテムの場合は当てはまらないことが多い
- ハザードを見つけるときのガイドワードを，エラー型を定める開始点とできる
  - アイテムスケッチとともにゴールモデルの記述に対して機械的にガイドワードを適用する（本発表外）．この両者には対応関係がある．幾つか例を示す

ガイドワード	関連エラー型
NO	ServiceOmission, ItemOmission
More	ItemCommission
Early	EarlyServiceStart, EarlyServiceTermination
Late	LateServiceStart, LateServiceTermination

まとめ

NIL

- 本発表では，ISO26262の概念段階において，機能安全仕様を記述するための手法について説明した．本論で解決する課題は次の二点である．

  （A）抽象的な機能安全機構
  （B）AADL記述によるフロー解析，エラー記述

- 前者については，入力検査と出力検査を含むモデルを提示した．後者は，AADLにより，各種時間間隔（緊急動作時間間隔およびFTTI）の算出方法を示した．ここでのモデルは，単純なケースにおいては，適宜簡略化することができる．エラーはハザードと関連しており，容易にAADLのエラーモデルに対応づけることができる