概念段階におけるハザード・脅威の識別手法

伊藤　昌夫[†1, †2]

本論文では，概念段階におけるハザードおよび脅威の識別手法を示している．システムの安全性およびセキュリティを確保するためである．ネットワーク接続された組み込み機器において，ソフトウェアの重要性が高まるとともに，安全性およびセキュリティ確保の重要性も増している．概念段階とは，システム要求仕様を記述するための開発初期段階を指す．この概念段階に適用可能なハザードおよび脅威の識別手法は，これまで存在しない．ハードウェア中心のシステムの場合，漸次的な変更が主となるため，その必要性が低かったためと考える．しかし，新しいソフトウェア中心のシステムが増え，その重要性が増すと共に，概念段階におけるハザードおよび脅威の識別手法が求められている．本論では，アイテムスケッチとゴールモデルを用いた要求の整理と，これらの記述を利用した，ハザードおよび脅威を見つけるための手順を示す．乗用車のための機能安全規格との対応も示した．手法の適用例としては，先進運転支援システム（ADAS，例えば [1]）に関係するシステムを用いた．但し，本論での議論は，広い範囲で適用可能と考えている．

Finding Hazards and Threats in Concept Phase

Masao Ito[†1, †2]

Abstract

In this paper, we propose an approach for finding hazards and threats. Those are relating to safety and security respectively, that become important in especially the software-intensive embedded system such as ADAS (Advanced Driver Assistance Systems). The definition of the concept phase is the process in which we consolidate the requirements and create the specification. There is no appropriate method that we can apply in this phase to find both hazards and threats. Because the hardware-centered system usually evolves gradually, but recent new software-intensive systems born out of scratch and need the method to analyze hazards and threats. In this paper, we mainly focus on finding hazards and threats to assure the system safety and security. We use the item sketch and goal model and apply the guide words. We use the standards and example of the automobile for the explanation purpose, but we believe that we can apply this method to various domains.

1. はじめに

ソフトウェアが重要な位置を占めるシステムが，ますます我々の日常生活と密接に関わるようになっている．また，IoT(Internet of Things) [2] という言葉に代表されるように，これらシステムは，ネットワークを介して相互に接続され

【脚注】
†1　株式会社ニルソフトウェア
†2　有限会社 VCAD ソリューションズ

つつある．従って，システムの安全性およびセキュリティが，これまで以上に重要になってきている．

特に，近年進歩が著しい先進運転支援システム（ADAS, Advanced Driver Assistance Systems）は，その代表である．ADAS は，駐車支援・衝突緩和ブレーキ・先行車追従などのシステムを包含した名称である（本論では，ADAS を手法の例として用いている）．ADAS は，運転者の支援や代行を行う．システムの役割は大きく，安全性に大きな影響を及ぼす可能性がある．また，効果的な ADAS を実現するために，車両同士，あるいは道路上のインフラシステムとネットワーク結合されつつあるため [3]，セキュリティへの対応も重要な課題となっている．

本論では，概念段階においてハザード・脅威を識別するための手法を示す．安全性・セキュリティを確保するためには，ハザード・脅威を識別することが開始点となる．安全性の確保とは，網羅的にハザードを識別し，それに対する処置を決めることである．セキュリティに関しても同様に脅威を識別し，その脅威に対する対処を決定することが必要である．

なお，本論の概念段階は，乗用車の機能安全規格である ISO 26262[4] の第三部に対応する．もちろん，車両に限らず全てのシステム開発において，概念段階は存在する．ISO 26262 規格は，現段階で，もっともよく概念段階を整理していると考えており，本論ではこの規格との対応についても考える．

信頼性解析のための手法として，FTA（Fault Tree Analysis）や FMEA（Failure Mode and Effect Analysis）などの既存の手法（ハザード分析全般に関し整理した書籍としては，例えば，[5]）がある．これら手法を，概念段階において使用することはできない．既存手法は，システムを対象とし，そのシステムが，明確に分解されていることを前提としているためである．

また，よく知られているように，信頼性と安全性・セキュリティは，異なる概念である．システムが高い信頼性を持っていたとしても，事故の時に安全側に移行しなければ，安全性が高いとはいえない．もちろん，概念段階を除けば，共通する部分も多い．我々の方法においても設計段階では，FTA や FMEA を用いる．

後述するように概念段階では，システムそのものではなく，その抽象概念である「アイテム」を対象とする．「アイテム」という新しい概念を用いる理由として，ソフトウェアとハードウェアの進化の速度が影響していると考えている．機械に代表されるハードウェアの場合，大きな構造の変化は少ないため，漸次的に良くしていくことになる．逆に，ソフトウェアの比重が高いシステムにおいては，新しいシステムを相対的に作りやすい．従って，ADAS に代表されるソフトウェアの比重が高いシステムが増えている今，概念段階における新しい手法が求められている，と考えている．

本論で示す手法は，2 つの良い特徴を持つ．

- システム開発における概念段階で利用可能である
- ハザードおよび脅威の識別を同時に行うことができる

全体の構成は，次の通りである．2 章で，解決すべき課題を整理する．3 章では，最初に手法の概要を説明する．次に，例を用いながら，具体的なハザード・脅威の識別手順について説明する．4 章では，既存の研究との比較を行う．最後に，5 章でまとめを行う．

2. 課題

本章では，概念段階におけるハザード・脅威識別を行うときの課題整理を行う．最初に，概念段階を定義し，用語「アイテム」について考える．次に，安全性とセキュリティの関係について考える．これら課題整理は，先に挙げた本手法の 2 つの特徴と対応している．

2.1 概念段階と「アイテム」

最初に，概念段階とは何かについて整理する．本論でいう概念段階は，ISO 26262 の第三部に示されている範囲に含まれる．特にハザードと脅威の識別に関しては，第三部の 5.4.1 から 7.4. までが，対応する箇所である（表 2 参照）[1]．

概念段階は，アイテムの定義から始まる．規格の中で，用語「アイテム」は，通常とは異なった意味を持っていることに注意が必要である．アイテムとは，システムの抽象表現である [2]．例えば，衝突緩和ブレーキシステムを取り上げる．アイテムは，特定の車に搭載される具体的なシステムではない．衝突緩和ブレーキシステムとは何かを考えて，機能や非機能を整理し，アイテム境界を定めた，衝突緩和ブレーキの本質である．

アイテムではなく，具体的なシステムからスタートすると，そのシステムの枠内で，安全性やセキュリティを考えるしかない．従って，対応する範囲も限定的となる可能性がある．アイテムを対象にするのは，開発の初期段階において，本質的な安全性やセキュリティを持つシステムを目指すためである，と考えることができる．

ADAS に代表される車両の情報システムは，これまでにない新しいシステムである．従って，システムの抽象概念であるアイテムを用い，概念段階で，本質的な安全性確保を図る．そこから段階的に詳細化していくことで，新規の場合であっても，安全なシステムとすることを狙いとしている．

以上より，概念段階での課題は，新しい用語であるシステムの抽象表現としての「アイテム」を，如何に表現するか，ということになる．

【脚注】

1 ISO 2626 は 10 部から構成されている．パートと項番を組み合わせて示す場合がある．例えば，第三部の 7.4.2 を 3-7.4.2 と記述する．

2 アイテムの定義は規格中に 2 種類存在する．第一部の用語定義のものと，第十部の定義である．ここでは，最後に発行された第十部の定義を用いる．「（具体的なシステム）は抽象表現であるアイテムから生成（現実化）したものである」．第一部の用語定義とは異なっているが，第十部の定義の方が，他の箇所との整合性は高い．

2.2 安全性とセキュリティ

安全性・セキュリティを，ともに「危害から＜対象＞を守ること」と考えると，両者を同一の枠組みで扱うことができる．＜対象＞が人の場合，安全とは，人を危害から守ることである．＜対象＞がアセット（資産）の場合，セキュリティとは，アセットを危害から守ることである[3]．

一方で，違いもある．安全性の場合，その関心は，設計・実装上の検討不足（信頼性）や構成品の故障である．それに対して，安全機構を組み込み，問題が生じたときにどのようにふるまうかを検討する．一方，セキュリティの場合は，悪意ある第三者を想定した上で，設計・実装上の検討不足（脆弱性）を除去することに，関心がある．安全性・セキュリティを脅かすハザード・脅威の識別において，この違いを考慮する必要がある．

どういう結果になれば問題と考えるか，ということに関しても違いがある．例えば，アセットに対する侵害は，セキュリティの場合，直ちに問題となる．安全性においては，アセットの十全性が破られ，人に危害が生じる可能性があった時，はじめて安全性の問題となる．

従って，ここでの課題は，ハザードと脅威の類似性と差異を考慮しつつ，両者を適切に識別するための方法を見つけることである．

なお，本論で扱うセキュリティは，情報空間上に限定している．セキュリティという言葉は，情報空間以外でも用いられる．自動車に関して例を挙げると，物理的に車両に侵入し，ECU（Electronic Control Unit）ボードをすり替える，或いは，OBD（On-Board Diagnostics）から情報を抜き出すといったこともセキュリティ上の問題である．しかし，本論では，物理的なセキュリティ上の侵犯を扱わない．あくまで情報空間に限定している．ちなみに，情報空間上でセキュリティが問題になった事例として次がある．キーレスエントリーシステムにおいて，キーの保持者の近傍から車両まで情報をリレーすることにより，キーを保持していないにもかかわらず車両を操作可能となったケースである[6]．ここでは，物理的なセキュリティ上の問題は生じていない．純粋な情報空間上の問題である．

3. プロセス

本論で示す手法には，次の 4 つのステップがある．

- アイテムスケッチを作成する (3.1)
- アイテムのゴールモデルを作成する．同時にアイテムスケッチも詳細化する (3.2)
- 各ゴール記述文にガイドワードを適用する (3.3)
- アイテムスケッチを利用してハザードおよび脅威を識別する (3.4)

アイテムスケッチでは，静的および動的なアイテムの定義を行う．ゴールモデルでは，アイテムが持つゴール（トップゴール）の詳細化を行う．

ゴールの詳細化に合わせて，アイテムの静的・動的表現も詳細化する．具体的には，ゴール木の各階層（抽象度レベル）に応じて，アイテムスケッチを用意する．なお，ゴール木とは，後述するゴールモデル中の木構造のことである．厳密には，非循環有向グラフとなるが，説明のために，本論では「ゴール木」という表現を用いる．

次に，各ゴールの記述文にガイドワードを適用する．

ガイドワード適用文とアイテムスケッチを利用して，最終的なハザードおよび脅威の識別が可能となる．

注意すべきは，ゴールおよびアイテムスケッチの抽象度に応じた記述を，それぞれ最後まで維持することである．即ち，抽象レベルに応じたゴールがあり，それに対応するアイテムスケッチが存在する．詳細（低い抽象度）のみをメンテナンスすることは，アイテムに対する理解や，再利用の観点から不利である．

3.1 アイテムスケッチ

アイテムスケッチは，アイテムの構造・ふるまい，およびアイテム境界を明確にするために用いる．アイテムの機能・非機能要求はゴールモデルが受け持つが，ゴールモデル中の各ゴール記述文を利用することで，ゴールとアイテムスケッチは，対応関係を，ムリなく維持することができる．

3.1.1 ゴール記述文とアイテムスケッチ表現

アイテムスケッチには，静的および動的表現がある．静的表現は，図式（例えば UML のクラス図，SysML[7] の内部ブロック図，或いは CATALYSIS アプローチ [8] の仕様型表現）により与えることができる．本論の例では UML を用いている．

また，以降では，CACC（Cooperative Adaptive Cruise Control）[9] を，ゴールモデルの題材としている．CACC とは，車間制御クルーズコントロールシステム (ACC，Adaptive Cruise Control system [10]) に，車車間通信を付加したものである．CACC では，ACC が持つイメージ等から得た情報に加えて，先行車両と通信することによって，先行車両の情報を取得する．先行車を運転している人のアクション（例えばブレーキ踏下情報）が，通信を介して直ちに伝わる．レーダ等による測距結果のみを用いる場合と比べ，より素早く後続車は反応できる．他車と通信するため，計算機や携帯電話同様に，セキュリティ上の懸念があり，本論では，例に用いている．

例えば，次の要求文を考える．

(S1)（自車は）先行車を認識する.

S1 のアイテムスケッチは，図 1 の (b) である．後述する

【脚注】

3 もともと，古いラテン語の securus は，両者の意味を持っている．現代においても例えば，ロマンス語系に属するフランス語では，sécurité は，安全もセキュリティも表す．
しかし，本論では，安全を脅かす危害の発生可能源のことを「ハザード」，セキュリティの場合は，アセットに対する危害の発生可能源を「脅威」として，区別している．

ゴールモデル中のゴール（図 1 の (a)）から，(b) の静的表現を得ることができる．ここでは，自車，先行車をクラスとして含むパッケージ「認識」を示している．

なお，この図において，「自車」というクラスを追加している．要求では，主語に相当する部分を省略することが多いため，必要に応じて補う必要がある．また，パッケージ名として，「認識」を用いている．本来，機能に相当する部分（ゴール記述文の述部）は，アクションとしての記載が望ましい．CATALYSIS であれば，仕様型中で，アクションとして記述することができ，より素直な表現とすることができる．

アイテムスケッチの動的表現は，有限状態機械図によって与えることができる．次の要求文を考える．

(S2) 先行車を認識したならば，追従中状態に移行する．見失った（ロスト）場合は，待機中状態に移行する（図 2 左上 , FSM_A).

(S3) スイッチを ON にすることで，CACC は ON（モード）になる．OFF にすることで，OFF（モード）になる (図 2 左下 , FSM_B).

状態と遷移のトリガが要求中に記載されていれば，容易に要求と状態遷移図の間で対応付けを行うことができる．

アイテムスケッチの静的・動的表現は，同一対象の 2 つの側面を示している．例えば，静的表現における先行車と自車の追従関係（図 1 の (b)）は，動的表現（図 2）の FSM_A における「追従中」状態での関係である．また，FSM_B におけるスイッチは，静的表現の一部として，表現することになる（図 4 参照）

3.1.2 開発カテゴリとアイテムスケッチ

ところで，概念段階を開始する時に，アイテムスケッチのどの抽象度レベルから始めるかは，開発カテゴリに従って決まる．開発カテゴリとは，対象のアイテムが，新規のアイテムか，それとも，すでに解析済みのアイテムかということである（[4] の 3-6.4.1 あるいは，表 2 参照）．

あるシステムを作成するときに，既存のアイテム記述が存在し，その一部を変更する必要がある場合，解析済みのアイテムスケッチを利用することができる．このとき，影響が生じるレベルから解析をスタートする．もちろんゴールモデルに対しても必要な変更を加える．

3.1.3 概念段階終了時のアイテムスケッチ

本論のスコープ外であるが，概念段階は，要求仕様書の完成により終了する（安全性に限れば，並行して作成する機能安全要求仕様書の完成である）．アイテムスケッチは，このとき，抽象ハードウェア・ソフトウェアアーキテクチャのベースとなる．

3.2 ゴールモデルとゴールの詳細化

ゴールモデルは，安全性およびセキュリティ確保に限らず，アイテムの機能・非機能要求を整理するために用いる．本論では，KAOS アプローチ [11] のゴールモデルを利用する．KAOS は，代表的なゴール指向要求分析手法の一つである．

ゴールモデルを用いて，最上位の要求を詳細化し，最終的に要求仕様を得ることになる． ここで用いる記法を，簡単に説明する．主たるノードは，ゴール・ソフトゴール・障害・解決・仕様である．ソフトゴールは，ゴールの達成基準が明確に示されないゴールである．障害は，ゴールの達成を妨げるノードであり（3.2.3 項），その解決は，解決ノードを用いて示す．仕様は，最終的に確定した要求であり，ゴールモデルの葉の部分に相当する．

上位のゴールと下位ゴールは，AND 洗練あるいは OR 洗練によって結合する．前者（AND 洗練）は，ゴールの分解である．逆に言えば，全ての下位ゴールを合わせたものが

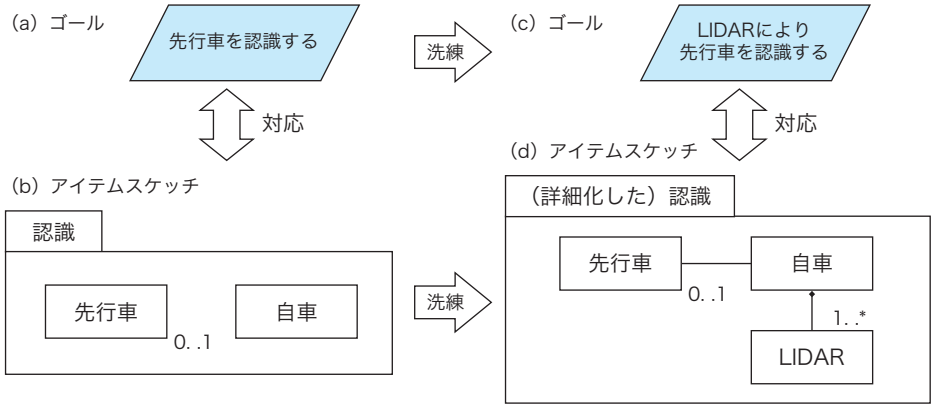

図 1　ゴールと対応するアイテムスケッチ（静的表現）

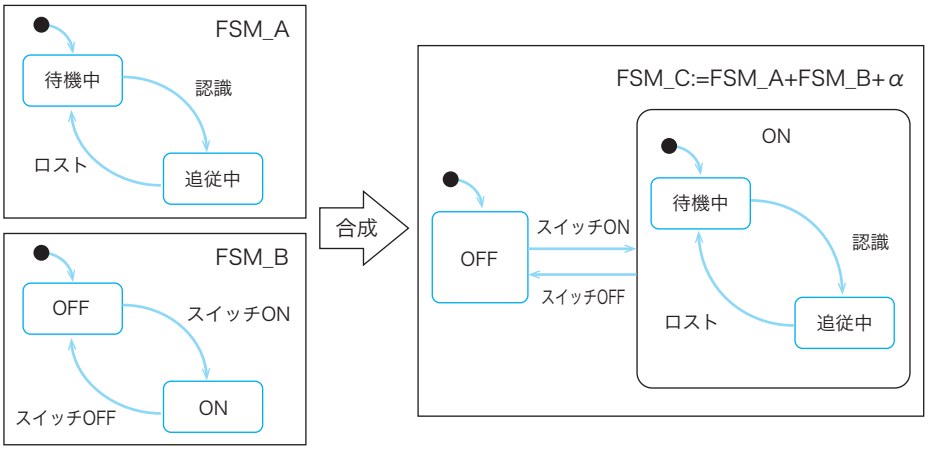

図 2　アイテムスケッチ（動的表現）とその合成

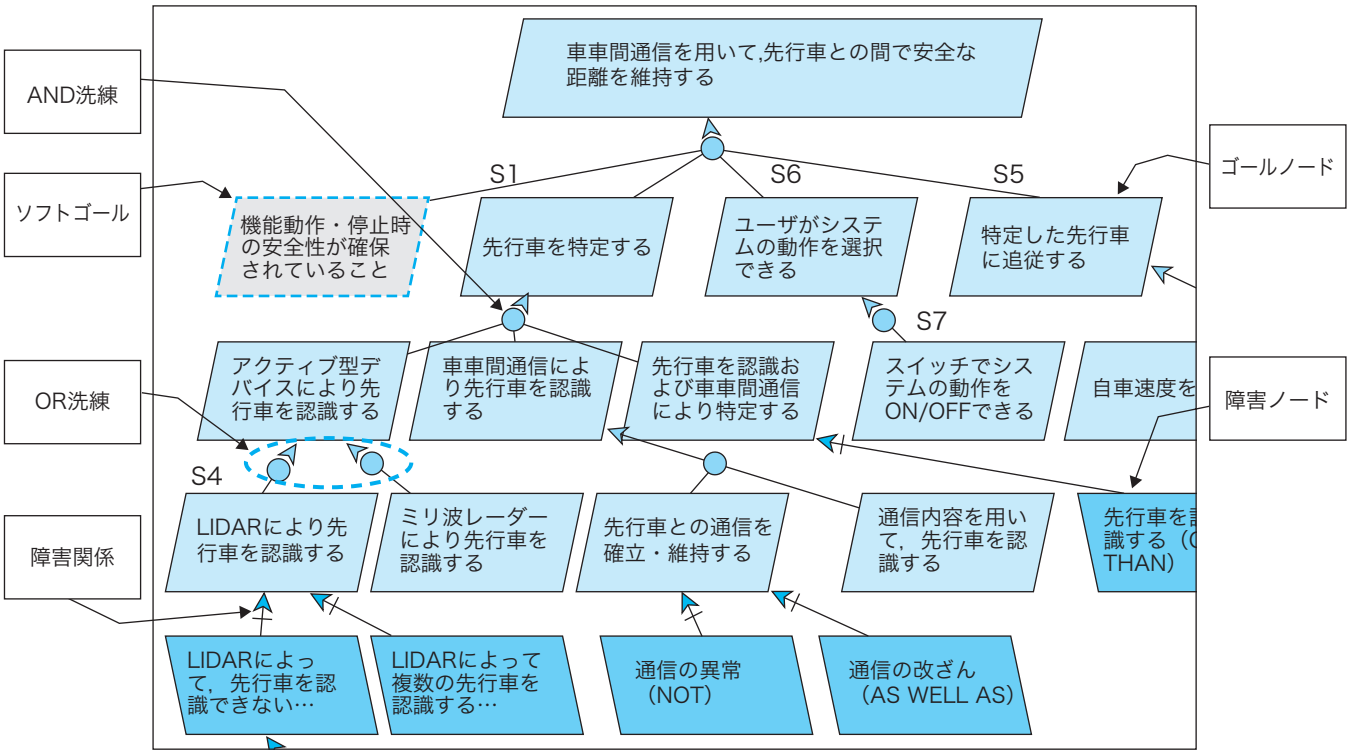

図 3　CACC のゴールモデルの例（本論に必要な範囲で単純化している，S1，S4，S5，S6，S7 は本文中のゴール記述文に対応している．但し，S1 は，一部変更している）

上位ゴールとなる．後者（OR 洗練）は，下位ゴールが上位ゴールの選択肢であることを示している．

CACC におけるゴールモデルの例を図 3 に示す．

次項からは，ゴールの詳細化に伴って，アイテムスケッチを組み合わせる二つの方法を示す．一つは，「洗練」であり（3.2.1 項），もう一つが「合成」である（3.2.2 項）．これらは，組み合わせから生じるハザード・脅威を識別するために用いる．

3.2.1 洗練

上位のゴールを，AND 洗練を利用し，複数のゴールに分割する．ゴールモデルをグラフとして見た場合，これは，下方向に枝を延ばすことに相当する．この時，ゴールと紐付いているアイテムスケッチをどう記述するが，本項の主題である．

次の例を考える．

(S4) LIDAR により，先行車を認識する

ここでは，アクティブ型デバイス，LIDAR[4] で先行車を認識することを示している．このアイテムスケッチは，図 1 右 (d) のモデルとなる．

図 3 に示すように，S1 と S4 の関係は，洗練関係にあり，基本的な構造を変えることなくアイテムスケッチを作成できる [5]．

3.2.2 合成

ここでの「合成」とは，直接の洗練関係にはないゴール同士を組み合わせることである．

アイテムスケッチの動的表現における例を示す．

次のゴール記述文について考える．

(S5) 特定した先行車に追従する

(S6) ユーザがシステムの動作を選択できる

(S7) スイッチで，システムの動作を ON/OFF できる

S5 は図 2 の左上（FSM_A）に対応し，S7 は図 2 の左下（FSM_B）に対応している．S6 は S7 の上位のゴールであるが，ここでは S5 と S7 で考える．なお，ゴールモデル上のそれぞれの位置は，図 3 参照のこと．ここまでで，2 つのアイテムスケッチの動的表現（FSM_A および FSM_B）を得た．この動的表現に対して合成を行う．前述のように「合成」とは，直接の洗練関係にはないゴール同士（即ち，異なる枝にいるゴール）を組み合わせることである．

なお，この場合，待機中・追従中は，あくまで ON 状態における状態遷移である．従って，ON 状態中の状態遷移と

【脚注】

4　LIDAR とは, Laser Imaging Detection and Ranging の略. 光を用いて，対象検知と測距を行う．前方の車両の認識技術としては，この他にミリ波レーダ・（赤外線）カメラなど様々なものがある．本論の例では，LIDAR をアクティブ型（認識用）デバイスとしている．認識技術が異なる場合は，バリアントとして，ゴールモデルを含む記述の再利用が可能である．例えば, 図 3 では, ミリ波レーダを OR 結合し, LIDAR とミリ波レーダは，代替の関係であることを示している．

5　図 3 のゴール木上では，S1 に相当するゴール記述は，「先行車を特定する」と述部の名称が異なっている．CACC では LIDAR 以外に通信を使用するためである．S1 を読み替えて頂ければ幸いである．

して合成する．先の「洗練」とは異なり，「合成」の場合は，（その記載が要求に含まれない場合）意味を考え組み合わせることになる．

3.2.3 障害ノード

障害ノードは，前述のようにゴールモデル中で用いるノードの一つである．ゴール達成の障害となる要素を表現する．例えば，「高い加速性能を有する」というゴールに対して，「運転者によっては，大きな加速度に不快感を持つ」は，障害ノードとなる．

この障害ノードを，本論では，ハザード・脅威の識別のために使用する．ハザード・脅威もまた，ゴールの達成を妨げるからである．次節では，ガイドワードを用いて，この障害ノードを如何に見つけるかを記述する．

3.3 ガイドワードの適用

各ゴールの記述文に対して HAZOP（Hazard And Operability Study）[12-15] のガイドワードを適用する．ハザードと脅威を見つけるための準備である．HAZOP ガイドワードは，2 つのタイプに分かれる．一つは，空間に関するもの．もう一つは，時間に関するものである（表 1 の次元欄を参照）．

表 1　HAZOP ガイドワード

ガイドワード	意味	次元
NO or NOT	否定	空間
MORE	量的な増加	
LESS	量的な減少	
AS WELL AS	質的な変化／増加	
PART OF	質的な変化／減少	
REVERSE	論理的に逆	
OTHER THAN	完全な代替	
EARLY	（時間的に）早い	時間
LATE	（時間的に）遅い	
BEFORE	（順番の）前	
AFTER	（順番の）後	

ガイドワードを用いた分析により，What-if（仮定）分析を，系統立てて行うことができる [16]．他の開発初期に用いられる手法とは異なり，チェックリストに依らず，対象を時空間上で網羅的に調べることができる．

今，ガイドワードを用いて，先のゴール S4 の変形を行う（日本語は，必要に応じて助詞等を変えている）．

（S4-NOT） LIDAR によって，先行車を認識できない

上記は，NOT ガイドワードを適用した場合である．ガイドワードによって，様々な吟味すべき文を作成することができる．いくつか例を挙げる．

（S4-MORE） LIDAR により，複数の先行車を認識する

（S4-ASWELLAS）LIDAR によって，誤って先行車を認識する

（S4-LATE） LIDAR による，先行車を認識したとの通知が遅れる

（S4-NOT_LATE）LIDAR による，先行車の認識不能との通知が遅れる

最後は，2 つのガイドワードを組み合わせた例である．

このようにゴール記述（S）に対して，ガイドワードを利用して，新しい文（S-*）を得ることができる．アスタリスクには，各ガイドワードが含まれる．即ち，一つの記述文から，複数の S-* を得ることができる．

次に，このガイドワードから生成した文を用いて，主題となるハザードおよび脅威の識別を行う．

3.4 ハザードおよび脅威の識別

ガイドワードにより生成した文（S-*）からハザードおよび脅威を識別する手順について説明する．

利用するのは，ここまでに既に作成した以下の記述である．

- アイテムスケッチ（3.1 項）
- ゴール記述に対して，ガイドワードを適用した文（S-*）（3.3 項）

「アイテムスケッチ上で操作を行うことで，ガイドワード適用文（S-*）を解釈する」というのが基本的なアイデアである．詳細は，次項で述べるが，アイテム定義上の記述や値を変更することで，何が生じるかを考えるという思考実験を行う．これは，ソフトウェアテストにおいてテストベクタ（入力の組）を見つけるために，境界値や境界値外に着目することに類似している．

最初にハザード識別に関して説明する．次に，脅威の識別に関して示す．

3.4.1 ハザード

単純なハザード識別としては，構成品の非正常動作をハザード候補として挙げる．アイテムを対象としているため，故障モードを単純に定めることは難しい，という点を除けば，FMEA に類似した方法である．例えば，S4-NOT:「LIDAR によって，（認識すべき）先行車を認識できない」では，アイテムスケッチの静的表現中の構成要素の故障モード（例えば，永続的な故障・一時的な故障，天候による検知不能）によりハザード候補を見つけることができる．

より複雑なハザードは，アイテムスケッチ中の情報（例えば，多重度）を操作することによって，見つけることができる．

例を挙げる．先行車は自車から見た場合，ゼロないしは 1 の多重度を持つ（図 1(b)）．いま，2 台以上の近接し同一速度で走行する先行車があったときに，（アイテムスケッチ上の記述ではゼロあるいは 1 なので）先行車を認識できない場合を想定できる．このように，存在する記述を削除する，或いは数を増やしてみることで，何が生じるかの思考

実験を行うことを，ここでは「アイテムスケッチを操作する」と呼んでいる．

別の解釈もできる．前述のケースで，システムは，（先行車はゼロ或いは 1 なので）誤って一台と見なすかもしれない．この場合は，S4-ASWELLAS（LIDAR によって，誤って先行車を認識する）と関係する．

ともに，静的なアイテムスケッチの多重度を「もし，先行車の多重度が 2 だったら？」と考え，各ガイドワード適用文（S-*）と関連づけることで，ハザード候補を得ることができる．

ゴールにガイドワードを適用した S-* から，どういう場合に問題が生じるかを，直ちに推定できるわけではない．上記に示したように，アイテムスケッチ上の情報の操作を通じて，その推定を行うことになる．

時間次元のガイドワードの例としては，S4-NOT_LATE がある．時間次元ガイドワードは動的表現を利用する．ここでは，図 2 を，参照のこと．先行車を見失った（ロスト）場合は，待機中に遷移する必要がある．しかし，先行車のロストを遅れて通知された場合，待機中に遷移すべきであるにも拘わらず，追従中のふるまいを続けることになる．先行車が急減速している場合は，危険な状態となる．この例では，ガイドワード適用文に相当する動的アイテムスケッチの該当箇所を探し，思考実験を行う．ガイドワード適用文のみを利用するより，より明確に危険な状況を理解することができる．

なお，LATE を適用したもう一つのガイドワード適用文 S4-LATE は，（使用性に影響を及ぼす可能性はあるが）安全性には影響しない．全てのガイドワード適用文が，ハザードに結びつくわけではないことに，注意が必要である．それでも，網羅性の観点から，問題がないという記録は必要である．

3.4.2 脅威

基本的な方法は，ハザードの場合と同等である．ゴール記述に対して，ガイドワードを適用することで，脅威の候補を見つけることができる．

但し，次の点で違いがある．ハザードは，アイテム（最終的にはシステム）の不十分な設計や，アイテムの要素であるエレメントの故障に由来する．一方で，脅威は，悪意ある攻撃者によるアセット（資産）への攻撃である．従って，ガイドワードを適用した場合の解釈を，変更する必要がある．例えば，S4-NOT は，「システムは第三者の攻撃によって先行車を認識できない」と解釈する．

ガイドワードを含んだ文（S-*）から脅威を見つけ出すためには，2 つの場所に着目する必要がある．最初は，脅威から保護されるべきアセットの場所であり，2 つ目は，他のアイテムとの相互コミュニケーションの場所である．

図 4 は，より詳細化したアイテムスケッチの版である．自車に付属する 3 つのクラスは，アセットを示している．これは，前述の着目場所のうち，前者（脅威から保護されるべきアセット）である．後者の相互コミュニケーションについては，通信デバイスを挙げることができる．これは車車間通信を行うもので，アイテム外（即ち先行車）との通信を行う．

ちなみに，更に詳細化を行うことで，異なる候補も現れる．アイテム内部のコミュニケーションに対する脅威である．例えば，CACC のメインの処理が動作する ECU と異なる ECU で，LIDAR が動作する時，通信路による内部コミュニケーションが存在する．概念段階の後半部分では，アイテムスケッチの静的表現を進化させた抽象ハードウェアアーキテクチャの記述を行う．内部コミュニケーションもこの段階では，考慮が必要になる．但し，本論のスコープ外となる．

さて，アセットのうち，ここでは，履歴データに着目する．履歴データの役割は，LIDAR と通信による先行車認識において，それぞれ時間軸上のデータ管理を行うこととする．先行車の安定的な認識を行うための仕組みである．即ち，通信対象の先行車が突然過去の値と（物理的にあり得ない）大きく異なる値を返したときは，異常とみなすために使用しているとする．履歴データが改ざんされると，存在しない先行車を存在すると見なす，或いは，逆に存在している先行車を存在しないと判断するかもしれない．従って，このアセットへの攻撃は，セキュリティばかりか安全性にも影響を及ぼす．

脅威の識別に関してまとめる．基本的な識別法は，ハザードと同様にゴールに対してガイドワードを適用することにより可能である．加えて，脅威とはアセットへの攻撃であり，アセットを識別することが重要である．これには，アイテムスケッチを利用する．アセットを意識することで，適切にガイドワードを含んだ文（S-*）から脅威を識別することができる．

3.5 SSM

ここでは，環境条件を定義するために考案した SSM（Situation-Scenario Mapping）について簡単に示す．車両のような移動体で，かつ車両レベルでハザード・セキュリティを考える場合，どのような環境の組み合わせがあり得るかを考えることは重要である．そのため SSM を記述する．（SSM はアイテムと相互作用するが，アイテムの一部ではないため，図 4 では，独立したクラスとして示している．具体的な環境からアイテムが受ける影響の例としては，LIDAR に対する天候・視程の影響がある）．

一般に，組み込みシステムの制御を考える場合，制御器（コントローラ）と制御対象（プラント）を用いてモデル化することが多い．しかし，環境と相互作用するシステムでは，環境もまた考慮すべき対象である．例えば，定速走行したいが，道路に勾配がある，或いは，天候によって路面の摩擦抵抗が変化しているときは，制御も影響を受ける．ここでは，環境に対して，状況を示す要素を選択的に与え，そ

の組み合わせをシナリオとした表を用いる．例えば，「＜晴天＞で＜高速道路＞を＜一定速度＞で走行する」というシナリオの場合，＜＞で示されるのが状況要素になる．

参考として，Appendix に，SSM の例を示す．

ISO 26262 が要求するリスク評定において，アイテムに ASIL を付与するときにも，シナリオは必要な情報であり，車両レベルで考える必要がある．ASIL は，ある環境中の車両に，故障が発生する発生する度合い・ドライバの対応可能性・事故が発生したときのドライバへの身体的影響から算出する．そのためには，具体的なシナリオが必要である．シナリオ毎に ASIL を計算し，想定可能な複数のシナリオからもっとも厳しい ASIL を付加することになる [17].

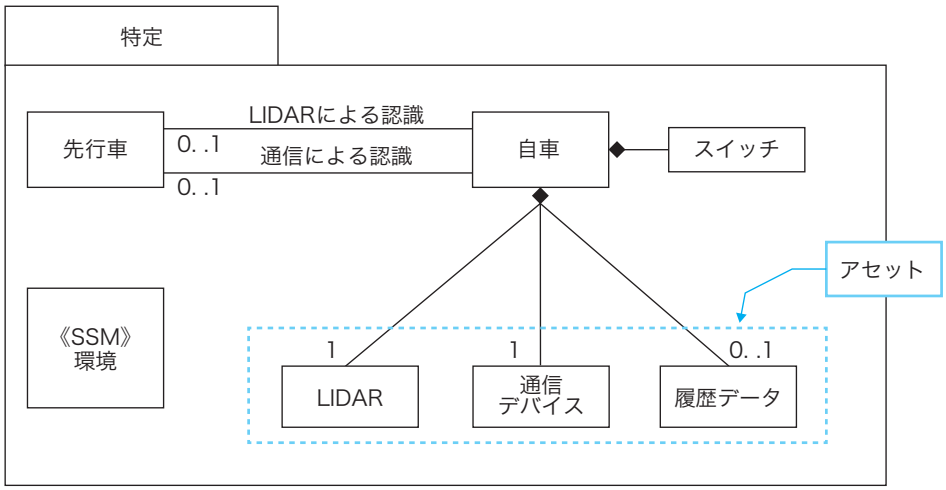

図４　詳細化したパッケージ「特定」とアセット

従って，ハザード識別に限らず，SSM は有効に利用できる．

システムは様々な環境で能動的に動作するため，最初の SSM 作成の負荷は大きい．しかし，一度作成すれば，同種のアイテムにおいては，再利用可能である．例えば，CACC 用に作成した SSM は ACC でも使用することができる．

なお，環境モデルである SSM および，今後ドライバ支援システムの増加とともに，必要性が増しているドライバモデルの作成手法については，別途詳細に記述したいと考えている．

3.6 規格との対比

ISO 26262 との対比を，表 2 に示す．本論で対象とするのは，アイテム定義，安全ライフサイクルの開始，ハザード分析とリスクアセスメントの一部である（3-7.4.2 まで）．表 2 では，この範囲に限定して「要求と推奨」の内容と本手法の関係を示している．

セキュリティに関して，同様の規格は，現在存在していない．しかし，いくつかの提案がなされている．例えば，[18] は，ISO 26262 に対応した脅威分析とリスクアセスメントを提案している．Severity（重大さ）や Controllability（制御可能性）については新たな定義を行っている．プロセスは，ISO 26262 と同一にできるとしている．

4.　関連研究

概念段階において適用可能なハザードおよび脅威の識別手法は，我々の知る限り存在しない．従って，ここでは，範囲を広げ，関連する研究および手法について，記述する．

初期段階の安全性解析手法として，PHA（Process Hazard Analysis），What-If 分析，HAZOP が知られている [16]．HAZOP 以外は，基本的にチェックリスト方式を用いる方法である．チェックリストにより網羅性を担保する．従って，如何にチェックリストを作るかが重要であるが，新しいシステムにおいて，網羅性を担保したチェックリストを作ることは困難である．本論のアプローチは，HAZOP のガイドワードを用い，更に，アイテムスケッチを組み合わせることで，チェックリストに頼ることなく，網羅性を確保している．

STAMP に基づく STPA のアプローチは，コントローラの相互作用に注目する解析手法である [19]．しかし，開始点として機能制御図等を使用するため，概念段階には使用することができない（[19] p. 213 “STPA uses a functional control diagram and the requirements, system hazards, and the safety constraints and safety requirements …”）．[20] では，STPA-sec としてセキュリティ拡張を行ったと主張している．STPA では，環境を陽に扱わないために，自動車のような動作環境が一定ではなく，かつ操作者も多様な場合には適用しづらい．我々の提案手法では，SSM によって環境を定義することができる．

セキュリティに関する総合的な手法としては，CORAS アプローチ [21, 22] がある．CORAS アプローチは，脅威図を中心として，解析を進める．しかし，概念段階における詳細な手順を持たない．

自動車分野に特化した方法としては，欧州の EVITA（E-safety Vehicle Intrusion proTected Applications）がある [23, 24]．EVITA は，特に車両内のコミュニケーションに特化したリスク分析を行う．この手法では，ユースケースからダークサイドシナリオを用いて，アセットを導出する．アセットの候補としては，性能・安全性・プライバシー・アセットがある．このアプローチは，概念段階終了後に，我々の手法と接続して使用することができる．

オリジナルの KAOS 手法を，セキュリティ問題に適用する方法についても報告がある [25]．ここでは，セキュリティゴールメタクラス（秘密性，保全性，可用性等）を特殊化することによって，網羅性を確保している．特殊化するときのパターンは，チェックリストのゴール表現と考えるこ

とができ，本手法と共に（検証のために）用いることも可能である．

5. 結論

本論では，概念段階におけるハザードおよび脅威の識別法について示した．アイテムスケッチとゴールモデルをベースとし，ガイドワードを利用することで，網羅的にハザードおよび脅威を識別することができる．具体的な例とともに手順を説明した．

また，本論に示した方法は，ISO 26262 の概念段階の要求事項と対応づけることができることも示した．

本論は，あくまでハザード・脅威の識別までがスコープである．しかし，本論での記述内容は，以降のフェーズでも利用することができる．例えば，ASIL 決定に必要なシナリオは，すでに SSM を用いて記述してある．また，解決ノードを障害ノードに対置することで，最終的に必要となる安全ゴールを定めることができる．

最後に，次の点を強調したい．本手法は，安全性・セキュリティ確保を，通常の要求分析プロセスと並行して行うことができるということである．本手法は，ゴール指向要求分析とともに用いている．トップゴールは，あくまでアイテムに対する要求である．それはゴールモデルの中で，詳細化され最終的には，要求仕様書になる．その過程で，各ゴールの評価を行うことで，安全性・セキュリティに関して担保できる．また，安全性とセキュリティが，必ずしも独立しているわけではないことは，すでに記した通りである．つまり，車両システムへの悪意ある侵入によって，安全性が脅かされる場合も考えられる．このとき，安全性とセキュリティを別々に解析していると見逃す可能性もある．

従って，要求分析を実施しながら，安全性およびセキュリティ確保のためのハザード・脅威の識別を同時に実施できることは，本手法の特徴であり，かつ最大の貢献と考えている．

謝辞

本論文および本論文の内容を口頭発表した第 11 回クリティカルソフトウェアワークショップでの匿名の査読者の方々および，欧州の ECQA（European Certification and Qualification Association）の機能安全コースにおいて，有意義なコメントをくださった各国の参加者のみなさんに感謝致します．

表 2　ISO 26262 との対比（概念段階のうち 3-7.4.2 までが本論文の対象）

ISO 26262（要求と推奨）				本論でのアプローチ
3-5	アイテム定義	3-5.4.1	アイテムの機能および非機能要求．アイテムの依存関係とその環境を含むこと	ゴールモデルを用いて，アイテムの機能および非機能要求を示す．また，アイテムスケッチは，このゴールモデルを理解するための静的・動的表現である．SSM は，アイテムが動作する環境を示す
		3-5.4.2	アイテム境界，アイテムインターフェイス，他のアイテムとの相互作用に関する仮定を定義すること	アイテムスケッチの静的表現により，アイテム境界やアイテムインターフェイス・他アイテムとの相互作用を定義することができる
3-6	安全ライフサイクルの開始	3-6.4.1	開発カテゴリを決定すること：新規開発か，既存アイテムの変更あるいは，動作する環境の変更か	開発カテゴリが，「既存アイテム」変更の場合は，変更が必要な抽象度レベルから開始することができる（本文 3.1.2 項「開発カテゴリとアイテムスケッチ」参照）
		3-6.4.2	開発カテゴリが修正の場合，影響分析および可能な修正ライフサイクルを決定すること	ゴールモデルでは，洗練関係が示されているため，容易に影響範囲を見つけることができる．即ち，変更の必要があるゴールに対して AND 或いは OR 洗練関係にある下位ゴールは，全て影響の有無を確認すべき候補である
3-7	ハザード分析とリスクアセスメント	3-7.4.1	アイテム定義に基づき，ハザード分析とリスクアセスメントを開始すること	アイテムスケッチとゴールモデルは，開始のために必要な材料となる
		3-7.4.2	状況分析とハザード識別を行う．動作状況とハザードの組み合わせから，ハザードイベントを決定すること	本論の主題である．状況分析については，本文 3.5 節が対応する

APPENDIX

SSM の例を示す．本文で例に使用した CACC における SSM となる．車載のシステム，例えば ACC は，この SSM を使用できる（状況のうち，「電波」が不要になる）．一方で，同じ車載のシステムである駐車支援システムでは，考慮すべき他の状況要素がある（例えば，有料駐車場におけるロック板の有無）

全く異なるシステム，例えば家庭内で移動可能なロボットにおいては，状況要素を改めて検討する必要がある．

しかし，それでもなお，枠組みとしては共通にできる．

[A] SSM

分類 / 時刻 (HH:MI:S)	道路				道路構造物		近傍車輌		道路混雑状況			気象・可視性			非自動車アクタ				法規			電波	タスク
	道路種*	道路状態*	車線数	曲率 (m)	照明	ガードレール他	前方車距離(m)	後方車距離(m)	進行方向	対向	交差*	天候*	気温 (摂氏)	視程 (度)	オートバイ	自転車	歩行者	障害物	区間内信号	速度制限 (KMH)	その他	到達可能距離 (m)	
1010:00	RT_SB	GR(0), GG(0), MU(0.8)	2	-	有	有	30	20	10	8	-	CM_CS	28	8	0	2	0	0	直進・青	M60	なし	200	T_FL
1012:00	↑	↑	↑	-	↑	↑	30	20	↑	↑	↑	↑	↑	↑	↑	1	↑	0	↑	↑	↑	↑	↑
…																							
1030:00	RR_CL	GR(0), GG(0), MU(0.6)	1	-	なし	なし	150	200	1	3	-	CM_CS	28	8	0	0	0	0	-	M40	なし	200	T_DR

*: 別表あり

区間あたりの台数

区間あたりの数

[B] 道路種

道路種	市街地	高速道路	自動車専用道路	幹線道路	バイパス	郊外道路	田舎道
RT_ID	DT	FW	LH	UA	BP	SB	CL

[C] 道路状態

道路状態	勾配	うねり	μ
RS_ID	GR(v)	GG(p)	MU(v)
	v:勾配%	p:パターン e.g. 波状路	v:摩擦係数

[D] 交差

交差する道路	交差	食い違い交差点	環状交差点	立体交差	合流
CF_ID	CF(iv)	SJ(iv)	RA(iv)	GS	JC
	iv:交差数	iv:交差数	iv:交差数		

[E] 天候

天候	晴れ	曇り	雨	雪	霧	霧	霧
CM_ID	CS	CL	RN(v)	SN(v)	HL	FG	HZ
			v: 降水量	v: 降雪量			

[F] タスク

タスク	タスク
T_SR	発車
T_DR	一定速走行
T_FL	先行車追従
T_CL	レーン変更
T_PS	追い越し
T_TL	左折する（交差点）
T_CL	左に曲がる
T_TR	右折する（交差点）
T_CR	右に曲がる
T_UG	急停止する
T_ST	（一時的な）停止する
T_PO	路肩にとめる
T_PK	駐車する
T_OB	障害物を避ける
・・・	・・・

図 5　CACC における SSM の例

[A] が SSM の本体になる. 他の表([B]～[F])は, [A] を記載するために必要な定義を行っている表である. 各カラムは環境を構成する要素(状況要素)になる. 一つのレコードが，ある時刻における状況を示している．時刻は，秒単位で記載している．この状況の組の時系列変化は，一つのシナリオとなる．即ち，一つの SSM の表が，一つのシナリオとなる.

【参考文献】

[1] Thalen, J., ADAS for the Car of the Future. 2006.

[2] Gubbi, J., et al., Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 2013. 29(7): p. 1645-1660.

[3] Delgrossi, L. and T. Zhang, Vehicle safety communications : protocols, security, and privacy. Information and communication technology series. 2012: Wiley. xxvi, 372 pages.

[4] ISO, ISO 26262. Road vehicles - Functional safety -, 2011, ISO.

[5] Ericson II, C.A., Hazard analysis Techniques for System Safety. 2005: John Wiley & Sons, Inc.

[6] Francillon, A., et al. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. in NDSS. 2011.

[7] OMG, OMG Systems Modeling Language (OMG SysML) V1.1, formal/2008-11-01, 2008, OMG.

[8] D'Souza, D.F. and A.C. Wills, Objects, Components, and Frameworks with UML: The Catalysis Approach. 1998: Addison-Wesley Professional.

[9] Naus, G., et al. Cooperative adaptive cruise control. in IEEE automotive engineering symposium Eindhoven, The Netherlands. 2009.

[10] SAEInternational, Adaptive Cruise Control (ACC) Operating Characteristics and User Interface (J2399), 2003, SAEInternational.

[11] van Lamsweerde, A., Requirements engineering: from system goals to UML models to software specifications. 2009: John Wiley & Sons Ltd.

[12] CEI/IEC, Hazard and operability studies (HAZOP studies) - Application guide, CEI/IEC 61882:2001, 2001, IEC.

[13] Fenelon, P. and B. Hebbron. Applying HAZOP to software engineering models. 1994. Citeseer.

[14] Nolan, D.P. and Knovel (Firm). Application of HAZOP and What-If safety reviews to the petroleum, petrochemical and chemical industries. 1994; viii, 128 p.].

[15] Redmill, F., M. Chudleigh, and J. Catmur, System Safety: HAZOP and Software HAZOP. 1999: John Wiley & Sons, Inc.

[16] Nolan, D.P., Safety and security review for the process industries : application of HAZOP, PHA, What-If and SVA reviews. 3rd ed. 2011, Oxford: Elsevier/GPP.

[17] Czerny, B.J., R. Suchala, and M. Runyon, A Scenario-Based Approach to Assess Exposure for ASIL Determination, 2014, SAE Technical Paper.

[18] Ward, D., I. Ibarra, and A. Ruddle, Threat Analysis and Risk Assessment in Automotive Cyber Security. SAE International Journal of Passenger Cars-Electronic and Electrical Systems, 2013. 6(2): p. 507-513.

[19] Leveson, N., Engineering a safer world: Systems thinking applied to safety. 2011: MIT Press.

[20] Young, W. and N.G. Leveson, An integrated approach to safety and security based on systems theory. Commun. ACM, 2014. 57(2): p. 31-35.

[21] Brændeland, G., et al., Using dependent CORAS diagrams to analyse mutual dependency, in Critical Information Infrastructures Security. 2008, Springer. p. 135-148.

[22] Lund, M.S., B. Solhaug, and K. Stølen, Model-driven risk analysis: the CORAS approach. 2011: Springer.

[23] Henniger, O., et al. Securing vehicular on-board it systems: The evita project. in VDI/VW Automotive Security Conference. 2009.

[24] Ruddle, A., et al., Security requirements for automotive on-board networks based on dark-side scenarios. EVITA Deliverable D2.3, EVITA project, 2009.

[25] van Lamsweerde, A. Elaborating security requirements by construction of intentional anti-models. in Proceedings of the 26th International Conference on Software Engineering. 2004. IEEE Computer Society.