演習コースIII

セーフティ&セキュリティ開発のための技術統合提案と事例作成

～STAMP/STPA とアシュアランスケースの統合～

A Proposal of a Technology Integration and Case Study for Development of Safety and Security

リーダー：大森　淳夫（パイオニア）　　中嶋　良秀（ノーリツ）
研 究 員：西村　伸吾（富士ゼロックス）　　久連石　圭（東芝）
柴引　涼（メタテクノ）　　邱　章傑（パナソニック）
久木元　豊（ﾃｯｸｽｴﾝｼﾞﾘｭｰｼｮﾝｽﾞ）　　松本　江里加（ダイキン工業）
荒井　文昭（ｷｬﾉﾝｲﾒｰｼﾞﾝｸﾞｼｽﾃﾑｽﾞ）　　細谷　雅樹（東光高岳）
神田　圭（日立ｿﾘｭｰｼｮﾝｽﾞ）　　太郎田　裕介（東京海上日動ｼｽﾃﾑｽﾞ）
主　査　：金子　朋子（情報セキュリティ大学院大学）
副主査　：高橋　雄志（トレドシステム）
アドバイザー：勅使河原　可海（東京電機大学）

研究概要

IoT 時代の開発方法論は，セーフティだけやセキュリティだけを意識したものではいけない．セーフティの考え方では，可用性を重要視するため，機器連携をする際に，情報の機密性が保たれていないことがある．一方セキュリティの考え方では，機密性を重要視するため，利便性や機能性を損なう可能性がある．IoT 時代を迎えるにあたって，これらのバランスの取れた開発方法論が必要である．しかしながら，バランスの取れた方法論は確立されておらず，既存のセーフティにおける開発手法や，セキュリティにおける開発手法がどの程度バランスの取れた設計手法として使えるのかの検証もされていない.本稿では，セーフティの分野で実績のある STAMP/STPA を,セキュリティの分野とコラボレートさせて，その有効性が検証できたので，セーフティ&セキュリティ開発のための方法論として提案するものである.

Abstract

The future of design methodology in Internet of Things (IoT) Era should not be conscious of only safety or security. On one hand, while focusing on the availability in terms of safety, the confidentiality of information may not be kept when linking equipment. On the other hand, although focusing on confidentiality is important from security viewpoint, the convenience and functionality may be impaired at the same time. As facing the IoT era, these balanced design methodologies are necessary. However, a well-balanced methodology has not been established, nor has it been verified whether development methods in existing safety or how to use the development method in security can be used as a balanced design method. In this paper, we have collaborated STAMP / STPA, which has been proven in the field of safety, with the field of security. Since its effectiveness has been verified, we propose it as a methodology for safety and security development.

1. はじめに

IoT（Internet of Things）デバイスの運用と AI 人工知能の発展は急激に普及し，異なる製品やサービスがインターネットを通じてつながり,新たなサービスや価値が提供され

る IoT 時代が実現しつつある．しかし,異なる製品やサービスがつながることで,IoT 機能を喪失すれば，人命に関わる範囲まで，安全性に影響を与える．また，IoT システムが被害に遭うだけではなく，IoT 機器が踏み台になった攻撃も増加している.「しかしセーフティとセキュリティ設計の必要性を認識しつつも，半数以上の企業では基本方針が設けられていない」など開発上の取り組みは不十分な状態にある[1]．本稿では，セーフティとは，偶発的なミス，故障などの悪意のない危険に対する安全を示し，セキュリティとは，悪意をもって行われる脅威に対しての安全を示すものとする．IoT 時代に求められる開発方法論は，セーフティまたはセキュリティの一方だけを意識したものではなく，両方を意識したものであるべきである.

セーフティとセキュリティの両立において，システムの安全性分析や脅威分析を行い，妥当性をもつリスク管理を実施する.そして,この根拠を示すことは非常に重要である．例えば，従来の解析手法である FTA（Fault Tree Analysis）や FMEA（Failure Mode and Effect Analysis）は単独のシステム・機器の分析をするには向いているが，人的要素を含めた分析が必要な自動運転等複数機器で構成しているシステムの分析には向いてないという指摘がある．そこで，安全性解析手法 STAMP/STPA（Systems-Theoretic Accident Model and Processes / System-Theoretic Process Analysis）が提案され，更に，セキュリティ面も含めた STPA-Sec も注目されている[2]．しかし，STPA は，セーフティとセキュリティ両方を考慮したハザード分析までであり，具体的な改善策に至っていないという課題がある[3].

本稿では，機器連携サービスの一例として自動車の自動運転サービスを取り上げて，セーフティの分野で実績のある STAMP/STPA を，STAMP/STPA-Sec，STRIDE と連携し，事例作成を行い，改良点や問題点の洗い出しを行った．また，分析結果を，アシュアランスケース[4]の手法の一つである CC-Case[5][6]を参考にし，保証のゴール，前提条件，戦略，手順を定めたうえで，脅威分析とリスク分析の検証，妥当性確認を行い，システムのセーフティとセキュリティの両立が検証できる一連の流れを提案する.

2. 関連技術

本章では，IoT 時代に適応可能な開発方法論として着目している既存技術を紹介し，どのように適応させようと考えているかを示す.

2.1. STAMP/STPA

2.1.1. STAMP/STPA の概要

STAMP とは，システム理論に基づく事故モデルのことであり，その STAMP アクシデントモデルを前提とし，システムのハザード要因を分析する新しい安全解析手法のことを STPA と呼ぶ[7].

STAMP/STPA では前提として，システム事故の多くは，構成要素の故障ではなく，システムの中で安全のための制御を行う制御要素と被制御要素の相互作用が働かない事によって起きるとしている．その前提を持って，要素（コンポーネント）と相互作用（CA：Control Action）に着目してメカニズムを説明し，アクションが働かない原因が CA の不適切な作用に等しいという視点を持つことで原因を有限化している [8].

以下に，STAMP/STPA の手順を示す.

・Step0：

前準備として，対象システムにおいて分析対象となる，アクシデント，アクシデントが潜在している具体的な状態であるハザードを定義し，ハザードを制御するためのシステム上の安全制約を識別する．その後，対象システムにおいて，安全制約の実現に関係するサブシステム，機器，組織等のコンポーネント，及び，コンポーネント間の CA，フィードバックデータといった相互作用を分析し，制御構造図（CS：Control Structure）を構築する

・Step1：

CS から、CA を識別し，4 種類のガイドワードを適用して，ハザードにつながる非安全な

CA（UCA：Unsafe Control Action）を抽出する
・Step2：
UCA ごとに，関係するコントローラーと被コントロールプロセスを識別して，コントロールループ図を作成し,ヒントワードを適用してハザード要因(HCF:Hazard Causal Factor)を特定する

2.1.2. STAMP/STPA-Sec

STAMP/STPA-Sec とは STAMP/STPA にセキュリティの要素を込みこんだ安全解析手法であり，従来のセキュリティ要求分析手法であるアタックツリーやミスユースケースのようにどのような脅威があるのかを洗い出す手段(How)ではなく,攻撃から何を守るべきか(What)を明確にするアプローチである[9][10].

本稿では，相互接続されたシステムに対し安全解析を行う手法として，セキュリティも考慮するために STAMP/STPA-Sec を用いる．ただし STPA-Sec 以外に STPA-SafeSec[11]も提案されており，STAMP によるセーフティ・セキュリティ分析の枠組みは確定していないため， Step2 で STPA-Sec のヒントワードに用いることに留めている．

2.2. STRIDE

STRIDE とはマイクロソフト社が定義する脅威モデルである．システムに対するセキュリティ上の脅威は様々なものがあるが，STRIDE では，Spoofing identity(なりすまし），Tampering(改ざん），Repudiation(否認），Information Disclosure(情報の暴露），Denial of Service(サービス不能），Elevation of Privilege(権限の昇格)という 6 つのカテゴリに分類している．名称は各カテゴリの頭文字を現したものである[8].

2.1.2 項の HCF を特定するために用いたヒントワードでは網羅性や必要性について十分ではないという問題がある[12]．本稿では，網羅性や必要十分な HCF が特定できるヒントワードを拡張するために，STRIDE を利用することを提案する．そして，その有効性を事例検証にて確認するものとする．

2.3. アシュアランスケースと GSN

2.3.1. アシュアランスケース

アシュアランスケース(Assurance case)とは，テスト結果や検証結果をエビデンスとしてそれらを根拠にシステムの安全性や信頼性を議論し，システムの認証者や利用者などに保証あるいは確信させるためのドキュメントである[13].

アシュアランスケースは，システムや製品の性質など証明したい主張に対して，説明，証拠，前提を用いて，主張の確からしさを説明する．そのため，アシュアランスケースには，構造と対象に，それぞれ最低限の要求がある．構造では，システムや製品の性質に対する主張,主張に対する系統的な議論,この議論を裏付ける証拠,明示的な前提が含まれ，対象では，議論の途中で補助的な主張を用いることにより，最上位の主張に対して，証拠や前提を階層的に結び付けることができる[14].

2.3.2. CC-Case

CC-Case とは，アシュアランスケースを拡張し，セキュリティ標準であるコモンクライテリア等のプロセスを組み合わせた開発方法論である[5].その構成要素の中に STAMP に基づくセーフティ・セキュリティ開発と，工程ごとのアシュアランスケースによる製品，システムの保証を掲げている[6].構成要素として論理モデルと具体モデルという 2 種類にアシュアランスケースがある．論理モデルとは，システム・機器を保証するための保証全体像を示す論理的プロセスである．各論理モデルのもとに具体モデルを展開する．具体モデルとは，そのシステム・製品ごとの具体的な特性をもったリスクへの検証をするアシュアランスケースである．本稿では，CC-Case の論理モデルと具体モデルに分ける考えを採用し，事例検証を行った．

2.3.3. GSN

GSN とは，欧州で約 10 年前から使用されているアシュアランスケースの代表的な表記方

法である[15]．前提とサブゴールに分けて，戦略を明示することにより論理関係を明確にした上で，最上位のゴールが成り立つことを保証する．本稿では，STAMP/STPA-Sec，STRIDE により脅威と対策を抽出した後に，システムの安全性や正当性を確認するため，表記法として GSN を使用することを提案する．

3. ケーススタディ

今回の例では，自動車の自動運転に焦点を当て，脅威分析から検証の一連の流れを作成した．まず，STAMP/STPA-Sec，STRIDE で脅威分析を実施後，対策を導出した．そして，GSN 表記に基づいて，結果を表現し検証した．自動運転技術の開発動向と技術課題[16]を参考に規定した自動車のシステムアーキテクチャを図 1 に示す．

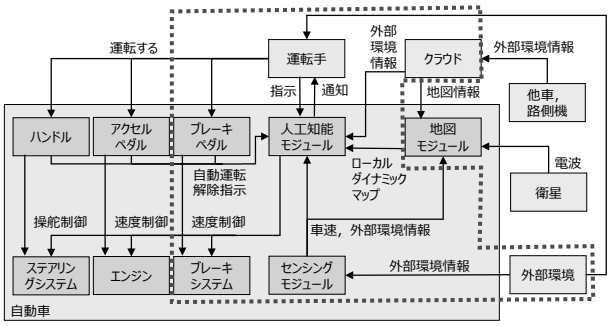

図 1　自動車のシステムアーキテクチャ

3.1. 手順

-手順1: アシュアランスケースによる保証全体像の決定

CC-Case の論理モデルに基づき，保証全体像を作成し，一連のプロセスを示す．プロセスの概略として，まず，セーフティとセキュリティの観点より分析する．次に，機能安全で規定している安全性のレベル（ハザード要因の発生頻度，回避可能性，過酷度）ごとに分類する．最後に，対策の評価と選択を行い，残存リスクを提示して，妥当性を確認する．また，この論理モデルをベースとして，本事例では,GSN を 2 階層に分割し，1 階層目を手順 5 で，2 階層目を手順 7 で，それぞれ紹介する．

-手順2: STAMP/STPA の Step0 を実施

手順 1 による絞り込みに基づき，CS の対象範囲の限定を行って CS の作成を実施．

-手順3: STAMP/STPA の Step1 を実施

-手順4: STAMP/STPA の Step2 を実施

本事例では，従来の Step2 に加え，STAMP/STPA-Sec と STRIDE によるヒントワードを拡張し，HCF の識別を行った．

-手順5: GSN を用いてハザードと UCA を整理

手順 2 で作成された STAMP/STPA の CS 図に対して，CC-Case の具体モデルに基づき，ブレーキ操作等に関する具体的なハザードと，STAMP/STPA の結果をつき合わせる．

-手順6: ハザード要因ごとに分析・対策立案

手順 5 で整理したハザード要因ごとに，ASIL（Automotive Safety Integrity Level）で分析する[17]．分析の結果，リスクが高いと判断した事象に対して対策を検討する．リスクが低いと判断した事象は残存リスクとして管理する．

-手順7: 対策妥当性の確認

手順 6 で検討した対策を，ハザードごとに GSN で論証する．

-手順8: 妥当性確認

手順 5, 7 で作成した GSN を統合し，手順 1 で決定した保証全体像に当てはめることによって，保証内容の妥当性を検証する．

3.2. 結果

-手順1: アシュアランスケースによる保証全体像の決定

保証する範囲を，人命・財産喪失という重大アクシデントに限定し，保証全体像を示す論理モデルとして，GSN を決定した.

-手順2: STAMP/STPA の Step0 を実施

識別したアクシデント，ハザード，安全制約を表 1，CS を図 2 に示す.

表 1 アクシデント，ハザード，安全制約

アクシデント (Loss)	ハザード (Hazard)	安全制約 (Safety Constraints)
(A1)自動車が外部環境(歩行者/他の車/周辺物)と衝突/接触する	(H1-1) 自動車が，ブレーキをかけても，外部環境の前で停止できない	(SC1-1) 自動車が，外部環境と衝突しないようにブレーキをかける(外部環境までの距離や相対速度を制御する)
	(H1-2)ブレーキがかからない	(SC1-2)運転手と自動車の両方がブレーキをかけられない状態にならない

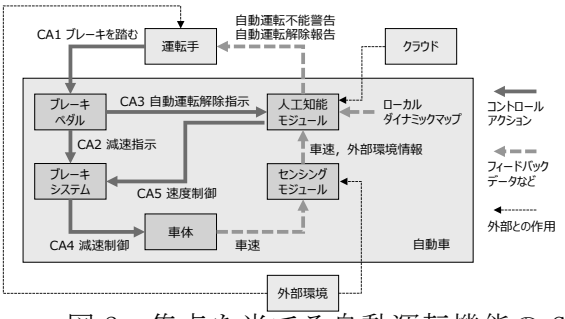

図 2　焦点を当てる自動運転機能の CS

-手順3: STAMP/STPA の Step1 を実施

手順 2 で識別した CS に基づき識別した UCA のうち運転手に関する CA1 とブレーキペダルに関する CA2 を表 2 に示す．この段階で，状況を明確にするコンテキストを UCA に追加した．例えば，UCA2 にコンテキストとして，ブレーキペダルの踏み込み度合いなどを追加した.

表 2　運転手とブレーキペダルに関する UCA

□コントロールアクション	Not Providing	Providing causes hazard	Too early / Too late	Stop too soon / Applying too long
CA1 運転手がブレーキを踏む	(UCA1-N) 運転手がブレーキを踏まないと危険回避ができず、外部環境と衝突する(SC1-2)違反	(UCA1-P) 運転手が誤った力加減でブレーキ操作を行うと、減速が弱く外部環境と衝突する (SC1-1)違反	(UCA1-T) 運転手のブレーキが遅すぎる場合、危険回避ができず、外部環境と衝突する(SC1-1)違反	(UCA1-S) 運転手がブレーキを踏む時間が短すぎる場合、危険回避ができず外部環境と衝突する (SC1-1)違反
CA2 減速指示	(UCA2-N) 運転手がブレーキペダルを踏んでいるのに減速指示を出さないと、外部環境と衝突する(SC1-2)違反	(UCA2-P) 運転手がブレーキペダルを強く踏んでいるのに減速指示が小さいと、外部環境と衝突する (SC1-1)違反	(UCA2-T) 運転手がブレーキペダルを踏んだタイミングに対し減速指示が遅すぎる場合、外部環境と衝突する (SC1-1)違反	(UCA2-S) 運転手がブレーキペダルを踏み続けているのに減速指示の解除が早すぎる場合、外部環境と衝突する(SC1-1)違反

-手順4: STAMP/STPA の Step2 を実施

手順 3 の結果より識別した HCF のうち特徴的な部分を表 3 に示す.手順 3 で UCA2 にコンテキストを追加したことによって，ブレーキペダルの踏込具合と減速指示の強弱が感覚的に不一致という具体的な HCF が抽出できた.

表 3　識別した HCF(抜粋)

UCA	(2)不適切なコントロールアルゴリズム	(T) Tampering 改ざん	(D) Denial of Service サービス不能
(UCA1-N) 運転手がブレーキを踏まないと危険回避ができず、外部環境と衝突する (SC1-2)違反	N/A	クラウドからの情報を改ざんし，人工知能モジュールに自動運転継続可能であると誤認識させる	人工知能モジュールに高負荷を与え自動運転不能警告を報知できない
(UCA2-P) 運転手がブレーキペダルを強く踏んでいるのに減速指示が小さいと、外部環境と衝突する (SC1-1)違反	ブレーキペダルの踏込具合と減速指示の強弱が感覚的に一致しない	N/A	N/A

-手順5: GSN を用いてハザードと UCA を整理

手順 2 から 4 の結果を，GSN を用いて整理した．自動運転におけるブレーキ操作に関するアクシデントをハザードと UCA により整理した結果を図 3 に示す．

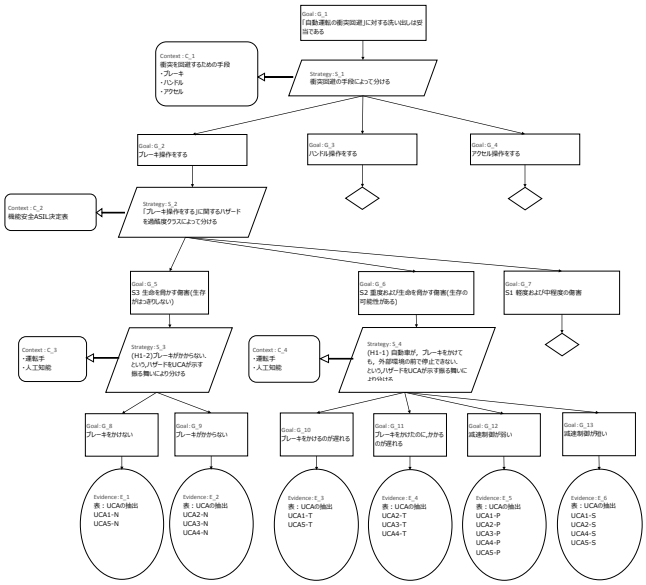

図 3　GSN によるハザードの整理

-手順6: ハザード要因ごとに分析・対策立案

手順 5 で作成した GSN に基づき，UCA ごとに整理したハザード要因に対して，ASIL 分析を行い，レベルごとに対策すべき HCF が整理できた．ブレーキ操作に関する HCF の ASIL 分析結果と対策の一部を表 4 に示す．

表 4　ブレーキ操作に関する HCF の ASIL 分析結果と対策

アクシデント	対象	該当する UCA	HCF	評価指標				対策内容	残存リスク
				過酷度クラス	発生頻度クラス	回避可能性クラス			
自動車が外部環境(歩行者/他の車/周辺物)と衝突/接触する	運転手－ブレーキペダル間	UCA1-N	悪天候など外部環境が悪く，運転手が危険察知しない	S3	E2	C2	ASIL_A	運転手の注意レベルを監視する	-
			運転手が危険を察知したが自動運転を過信して，ブレーキを踏まない	S3	E4	C2	**ASIL_C**	運転手の注意レベルを監視する 定期的に音声による注意喚起	-
			ブレーキペダルの遊びと認知する値が大きすぎて，ブレーキを踏んだと認識しない	S3	E2	C2	ASIL_A	ユーザビリティ評価を実施し，適切な遊び量に調整する	-

-手順7: 対策妥当性の確認

図 4 に示すような GSN をハザードごとに作成した．

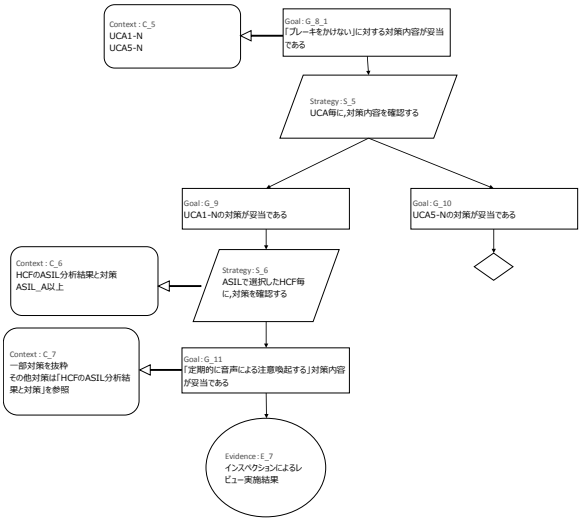

図 4　ブレーキをかけないハザードの GSN

-手順8: 妥当性確認

手順 1 で決定した自動運転に対するセーフティ&セキュリティ設計は妥当であるというゴールについて，ブレーキを操作するという手段を揺るがすハザードに対して，対策と残存リスクが妥当であることを確認できた.

3.3. 考察

本事例では，従来セーフティの手法である STAMP/STPA に STRIDE をヒントワードとして拡張することで,セーフティとセキュリティの脅威を洗い出すことができた.そして,STAMP の特徴である要素間の相互作用をモデリングすることによって，運転手と自動車の両方に関わる要因を考慮できた．情報の改ざんや DoS 攻撃によるシステムダウンといったセキュリティ要因と運転手の危険察知の遅れやブレーキペダルの欠陥といったセーフティ要因が，今回の事例では該当する．これらの要因はシステムの安全性とセキュリティ確保を設計時に考慮する場合，有用であると考える．特に，STAMP/STPA のプロセスを複数回繰り返すことや Step の手戻りを実施することでドメイン知識が無くともモデル化できるメリットがあると言える.

本事例では，アクシデントを人命喪失に限定したが，情報漏洩などのセキュリティ要因による損失をアクシデントと識別することで，セーフティ以外にセキュリティにも対応できると考える.

従来の STAMP/STPA Step1 の UCA にコンテキストはない．しかし，本事例では，状況を明確にするコンテキストを UCA に追加することにより，具体的な HCF を抽出することが可能となった.

STPA-Sec で追加したヒントワードでは，HCF を 1 件しか抽出できなかったが，STRIDE をヒントワードとして拡張することで，HCF を 33 件抽出できた．特になりすましや権限の昇格から導出した HCF は権限設計に活かすことができると考える．また，CS にデータの流れやデータストアを記載することによって，STRIDE による HCF 抽出が容易になると考える.

STAMP/STPA を用いた分析は,Step0 から 2 まで工程を繰り返すことで洗練された CS を作成できることがわかった．本事例では，ハザードの設定を 2 回，CS の設定を 3 回繰り返した.

本事例では，複数人で分析を行った結果，アクシデント，ハザード，UCA，HCF の整合性や粒度が異なった．分析者の意識統一のため，これらの用語について，当該ドメインの用語に読み替えた具体例を事前に定義することが必要であると考える．例えば，アクシデントやハザードの前提条件，HCF のヒントワード等についてである．ただし，多様性を確保するために既存の定義やヒントワードは残しておくべきである.

最後に，アシュアランスケースの考え方で情報を整理することで，STAMP/STPA より得られた結果をそのまま分析する前に，ハザードを整理することができた．そして，GSN でハザードごとに対策を検証することで，対策の妥当性が確認できた.

4. 今後の課題

本稿では，CS を従来の表記法に準ずる形で作成を開始した．しかし，データストアやデータの流れの記載に関するルールがなく，なりすましや情報の改ざんに関する HCF の洗い出しが困難であるという課題を発見した．そのため，データストアやデータの流れを記載するルールを追加することを今後の課題とする．その結果，STAMP によるセーフティ・セキュリティ分析の枠組みは確定できるようになると考える.

また，本稿のように複数人で分析する場合は，各種整合性や粒度が異なるといった課題がある．分析者の意識統一を図るため，前提条件やヒントワードの追加を検討したい.

5. まとめ

本稿では，機器連携サービスの一例として自動車の自動運転システムを取り上げ，事例

作成を通してセーフティとセキュリティの両立した設計の検証・妥当性確認を行った．セーフティの手法である STAMP/STPA（STPA-Sec）に STRIDE のヒントワードを拡張することで，セーフティとセキュリティ両方に関する脅威を洗い出すことができた．そして，STAMP の特徴である要素間の相互作用をモデリングすることによって，運転手と自動車の両方に関わる要因を考慮した分析結果を得た．また，アシュアランスケースの手法で脅威分析とリスク分析を実施した結果，GSN により図式化した理解しやすい対策を提示できた．

今後，4 章で述べた課題に取り組むと共に，更なる事例作成，普及展開を図る．

参考文献

[1] IPA/SEC，セーフティ設計・セキュリティ設計に関する実態調査結果，2015
[2] Haruka Nakao, Masa Katahira, Yuko Miyamoto, Nancy Leveson, Safety Guided Design of Crew Return Vehicle in Concept Design Phase Using STAMP/STPA, Proceedings of the 5th IAASS Conference A Safer Space for Safer World,2012
[3] 八山 幸司，米国における STAMP（システム理論に基づく事故モデル）研究の最新の動向，JETRO/IPA NewYork，2015
[4] T. P. Kelly, Arguing Safety - A Systematic Approach to Safety Case Management, DPhil Thesis YCST99-05, Department of Computer Science, University of York, UK, 1998.
[5] 金子朋子，山本修一郎，田中英彦，CC-Case～コモンクライテリア準拠のアシュアランスケースによるセキュリティ要求分析・保証の統合手法，情報処理学会論文誌 55 巻 9 号(2014)
[6] 金子朋子，髙橋雄志，勅使河原可海，吉岡信和，山本修一郎，大久保隆夫,田中英彦，セキュリティ要求分析・保証の統合手法 CC-Case の有効性評価実験，情報処理学会論文誌 コンシューマ・デバイス&システム(CDS) Vol.8 No.1，pp.11-26，2018.1
[7] システム安全性解析手法 WG，はじめての STAMP/STPA～システム思考に基づく新しい安全性解析手法～，Ver1.0，2016.3
[8] 金子朋子・高橋雄志・大久保隆夫・勅使河原可海・佐々木良一，安全解析手法 STAMP/STPA に対するセキュリティ視点からの脅威分析の拡張提案，Computer Security Symposium 2017，pp.1273-1279，2017.10
[9] システム安全性解析手法 WG，はじめての STAMP/STPA（実践編）～システム思考に基づく新しい安全性解析手法～，Ver1.0，2017.3
[10] William Young Jr，Security Tutorial Part 1　A Systems Approach to Security，5th STAMP Workshop in BOSTON
[11] Ivo Friedberg, McLaughlin, Paul Smith, David Laverty, Sakir SezerKieran, STPA-SafeSec: Safety and security analysis for cyber-physical systems, Journal of Information Security and Applications, 2017
[12] William Young, Nancy Leveson. Systems Thinking for Safety and Security, Proceedings of the 29th Annual Computer Security Applications Conference (ACSAC 2013), pp.1-8, 2013
[13] 松野裕,高井利憲,山本修一郎，D-Case 入門～ディペンダビリティ・ケースを書いてみよう！～，2012
[14] 金子朋子，セキュリティ・バイ・デザインとアシュアランスケース，SEC journal Vol.12, No.3，28-33，2016
[15] Tim Kelly and Rob Weaver, The Goal Structuring Notation - Safety Argument Notation, Proceedings of the Dependable System and Networks 2004 Workshop on Assurance cases, 2004
[16] 須田 義大，青木 啓二，自動運転技術の開発動向と技術課題，情報管理，57 巻 11 号 p.809-817，2015
[17] 茂野 一彦，自動車用機能安全規格 ISO26262 の紹介，MSS 技法・Vol.23，pp.23-38，2013