システム理論に基づくアクシデントモデルSTAMP

SEC調査役 石井 正悟

IoT（Internet of Things）の普及に見られるように、コンピューターで制御されたシステムは、日常生活のあらゆる場面において密接に関与するようになってきて、システムの安全性向上が大きな課題になっている。一方で、システムの高機能化に伴い個々のシステム自体が大規模・複雑になり、更に様々なシステムがネットワークによって相互に接続されてシステムの大規模・複雑化に拍車がかかっているため、システム全体の安全性確保がますます難しくなってきている。実際に近年は、システム障害（アクシデント）もシステム構成要素に起因するのみならず、構成要素同士の間、ないし、システムと人間との間の複雑な相互作用によるものがしばしば発生している。その背景には、従来の安全性の考え方及びシステム開発手法は「アクシデントはシステム構成要素の故障に起因する」と仮定しており、システム構成要素の故障以外の事故原因（ハザード要因）をカバーしきれていないことが挙げられる。このような状況においてIPA/SECでは、システムの安全性に関して世界的に著名な米国マサチューセッツ工科大学（MIT）のNancy Leveson教授が提唱しているSTAMP（Systems-Theoretic Accident Model and Processes）に注目しつつ、コンピューターシステムの安全性向上に資する新たな手法について調査・研究・普及の活動を行っている。

1 STAMPとは

20世紀に開発されたシステムの多くは、構成要素が少なく、それらの役割も明確であり、それゆえアクシデントが起きた場合は、構成要素のどれが根本原因でアクシデントに至ったのかを分析することは容易であった。従来アクシデントは、根本原因となる機器の故障や人間のオペレーションミスがまず発生し、それがシステム内のほかの機器や人間に伝搬し、システム内で悪影響を食い止めることができずに、最終的に起こってしまうという、チェーンオブイベントの形のモデルとして捉えられていた。

しかし、21世紀になると、開発されるシステムの規模は非常に大きくなり、構成要素も爆発的に増大すると共に、要素間の相互作用も複雑になり、個々の要素の役割を理解しただけでは、もはやシステムを理解できなくなった。そのような相互作用が複雑なシステムにおけるアクシデントの原因は、一つの構成要素に限定できる要因だけではなく、複数の要素間の相互作用による要因も考える必要があった。

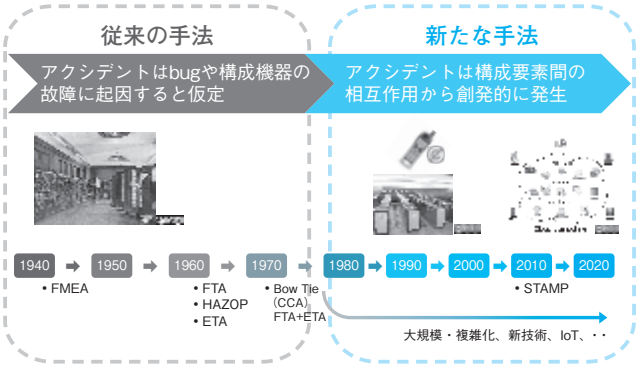

図1 コンピューターシステムと安全分析手法の変遷

MITのLeveson教授は、著書「Engineering a Safer World※1」の中で、システムの安全性は構成要素の相互作用から創発される（局所的な相互作用に隠れていたものが表面化して全体に影響を与える）ものであり、個々の要素を分割して分析するべきではないと述べた。そして、現代のシステムのアクシデントの多くは、システム構成要素の故障によって起きるのではなく、システムの中で安全のための制御を行う要素（コントローラー：Controller）と制御される要素（被コントロールプロセス：Controlled Process）の相互作用が働かないことによって起きるというアクシデントモデルを提唱した。このモデルを「STAMP（Systems-Theoretic Accident Model and Processes）：システム理論に基づくアクシデントモデル」と呼ぶ。

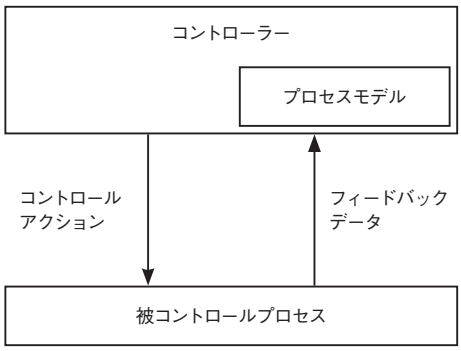

図2 STAMPにおける相互作用のモデル

STAMPモデルでは、システムの様々な階層でコントローラーと被コントロールプロセスに該当する要素が存在しており、それらの相互作用が適切に働くことによりシステムの安全が実現されるとする。STAMPモデルにおいて、アクシデントは相互作用が適切に働かないことによって起こり、具体的にはコントローラーから被コントロールプロセスへの必要な制御指示（コントロールアクション：Control Action）が適切に与えられないために起こるとしている。そして、不適切なコントロールアクションが与えられる要因として、コントローラー自身が想定する被コントロールプロセスの状態（プロセスモデル：Process Model）が、実際の被コントロールプロセスの状態を正しく反映できていないことが主要な要因であるとしている。たとえコントローラーも被コントロールプロセスも故障せずに、仕様通りに正しく動作していても、このような認識の不整合により不適切なコントロールアクションが与えられ、最終的にアクシデントにつながるというアクシデントモデルである。

2 安全性解析手法STPA

STAMP自身は分析手法ではなく、アクシデントを説明するモデルである。STAMPをベースとする、解析の道具立てやプロセスが幾つか提案されており、STPAはその一つでシステム開発を行う際などに用いるハザード分析手法である。

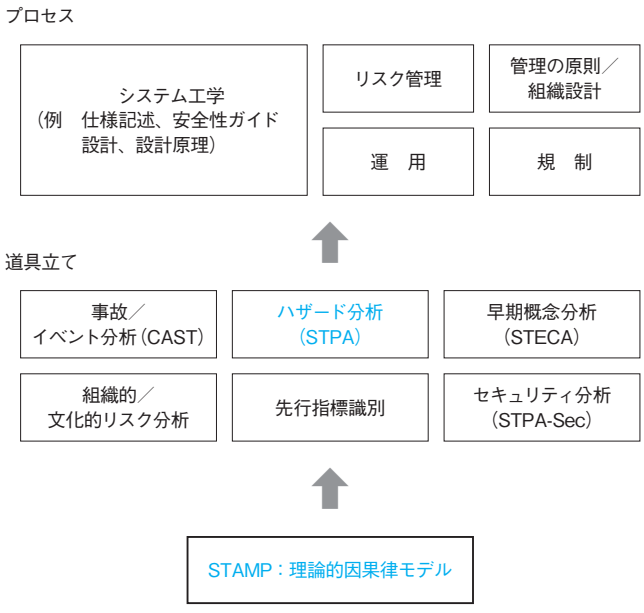

図3　STAMPに基づく分析の道具立てとプロセス※2

従来からハザード分析手法としては、強力でデファクトスタンダードとも言える手法のFTA（Fault Tree Analysis）、FMEA（Failure Mode and Effect Analysis）、HAZOP（HAZard and OPerability study）がある。それら従来手法とSTPAを比較すると表1のようになる。STPAは、複雑なシステムの「ソフトウェアの要求・設計ミス」を識別するのに適したものとされている。

表1　従来手法との比較

手法名	分析方法	特徴
従来手法（FTA、FMEA）	●フォールトツリー図や影響分析表を用いてハザード要因を分析する ●システムの構成要素と故障モードが決まるアーキテクチャー設計の段階から適用できる	●機器や組織の単一故障をハザード要因として識別する ●分岐条件を論理的に組むことで網羅的に分析できる ●深く分析できる反面、全体的な視野での分析が難しい
STAMP/STPA	●コントロールストラクチャーとコントロールループ図を用いてハザード要因を分析する ●システムの大まかな構成要素が決まる概念設計の段階から適用できる	●複数の機器や組織（人間）が、相互作用を行う複雑なシステムにおいて、相互作用のハザード要因を識別する ●過去のアクシデント事例データに基づくガイドワードにより網羅的に分析できる ●システム全体の振る舞いを確認しながら分析ができる

3 STPAの手順

STPAでは、Step 0（前準備）でシステムが安全を実現する大まかな構造を分析した後、Step 1でハザードに至るシナリオを、Step 2でハザードの詳細要因を分析する。

Step 0：（準備1）アクシデント、ハザード、安全制約の識別

対象システムにおいて分析対象となる、アクシデント、ハザード（アクシデントが潜在している具体的な状態）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。

Step 0：（準備2）コントロールストラクチャーの構築

システムにおいて、安全制約の実現に関係するコンポーネント（サブシステム、機器、組織等）、及び、コンポーネント間の相互作用（コントロールアクション、フィードバックデータ）を分析し、制御構造図（コントロールストラクチャー図：Control Structure Diagram。以下、コントロールストラクチャー）を構築する。

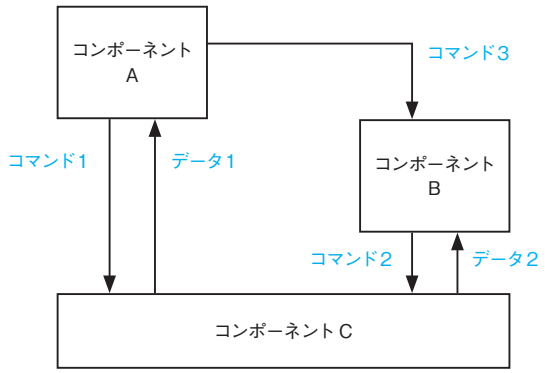

図4　コントロールストラクチャーの例

【脚注】

※1　Nancy G. Leveson：Engineering a Safer World: Systems Thinking Applied to Safety（Engineering Systems）, The MIT Press, 2012.

※2　An STPA Primer, http://psas.scripts.mit.edu/home/wp-content/uploads/2015/06/STPA-Primer-v1.pdf

Step 1：非安全なコントロールアクション（UCA）の抽出

コントロールストラクチャーから、安全制約の実行に必要なコントローラーによる指示すなわちコントロールアクションを識別し、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コントロールアクション（Unsafe Control Action：UCA）を抽出する。

表2　非安全なコントロールアクションの抽出例

コントロールアクション	非安全なコントロールアクション			
	与えられないとハザード	与えられるとハザード	早過ぎ、遅過ぎ、誤順序でハザード	早過ぎる停止、長過ぎる適用でハザード
コマンド1	XX条件下で、コマンド1が提供されない場合、ハザードに至る（UCA1）	コマンド1の内容が誤っていた場合、処理は停止するが、ハザードには至らない	コマンド1の提供が、コマンド2よりも遅れた場合、指示が上書きされ、ハザードに至る（UCA2）	コマンド1が途中で停止した場合、ハザードには至らない
コマンド2	コマンド2が提供されない場合、処理が開始しないが、ハザードには至らない	コマンド2の内容が誤っていた場合、処理は停止するが、ハザードには至らない	コマンド2の提供が、連続した場合、指示が累積され、ハザードに至る（UCA3）	コマンド2が途中で停止した場合、ハザードには至らない
コマンド3	コマンド3が提供されない場合、処理が開始しないが、ハザードには至らない	コマンド3の内容が誤っていた場合、処理は停止するが、ハザードには至らない	コマンド3の提供が、遅い／早い場合、処理開始がずれるが、ハザードには至らない	コマンド3が途中で停止した場合、ハザードには至らない

Step 2：ハザード要因（HCF）の特定

Step 1で抽出した非安全なコントロールアクションごとに、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、ガイドワードを適用してハザード要因（Hazard Causal Factor：HCF）を特定する。とくに、ソフトウェアや人間に起因する要因として、コントローラーの想定するプロセスモデルが、実際のプロセスの状態と矛盾することで起きる要因を特定する。

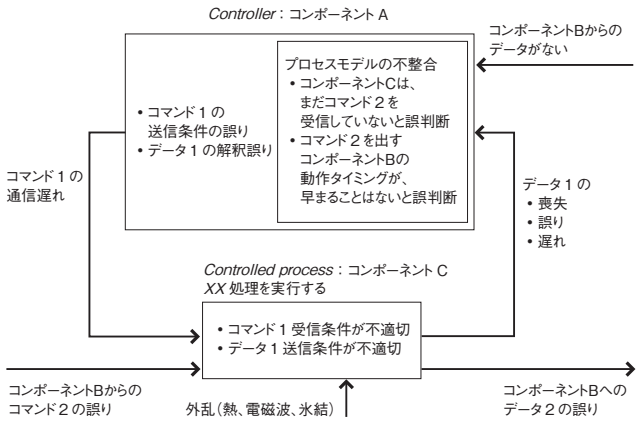

図5　ハザード要因（HCF）の抽出例

Step 1のUCA抽出において、想定外を排除するためのヒントとなるガイドワードが与えられている。

1. （**与えられないとハザード：Not Providing**）安全のために必要とされるコントロールアクションが与えられないことがハザードにつながる。
2. （**与えられるとハザード：Providing causes hazard**）ハザードにつながる非安全なコントロールアクションが与えられる。
3. （**早過ぎ、遅過ぎ、誤順序でハザード：Too early/too late, wrong order causes hazard**）安全のためのものであり得るコントロールアクションが、早過ぎて、遅過ぎて、または順序通りに与えられないことでハザードにつながる。
4. （**早過ぎる停止、長過ぎる適用でハザード：Stopping too soon/applying too long causes hazard**）（連続的、または非離散的なコントロールアクションにおいて）安全のためのコントロールアクションの停止が早過ぎる、若しくは適用が長過ぎることがハザードにつながる。

4 「はじめてのSTAMP/STPA」を公開※3

STAMP及びSTPAは非常に有効であると期待できるものの、既存の教科書は考え方を解説するものであるため、STAMP初心者にとっては理解が容易ではない。しかも、最新版を日本語に翻訳したものが公開されていない。更に、手法の実施手順についても前節のような概念的な説明であり、分析実施事例紹介でも分析実施結果のみの提示がほとんどである。そのため、いざ実際のシステムに対して分析しようとすると、どの開発ドキュメントの何と何を参照して、どのように作業すれば良いのか分からず、悩むことになる。あるいは、分かったつもりになって、誤った分析をしてしまい時間を無駄に経過させることになってしまいがちである。実際、IPA/SECがSTPAトライアルを実施したときにも上記の状況に陥った。

そこで、IPA/SECではSTAMP導入者向けに簡潔かつ具体的に手順を示す解説書を発行することとし、この度発行したのが「はじめてのSTAMP/STPA」という小冊子である。

IPA/SECでは、2015年度にシステム安全性解析手法WGを設置した。当WGでは、2015年6月にSEC特別セミナーに招聘したLeveson教授と意見交換会を実施し、それを受け我が国で実際に運用されている鉄道の踏切制御システムを対象に、STAMP有識者及び鉄道有識者を交えて、安全分析を実施した。その後2016年1月の13thWOCS[2]にLeveson教授を招聘した際にも意見交換会を実施し、上記分析結果を紹介したところ、Leveson教授から良好な評価を受けた。

本小冊子には、上記の分析結果を用いてSTPAの手順を具体的に解説している。

4.1 手順解説書の構成

本小冊子の構成は次のようになっている。

表3 「はじめてのSTAMP/STPA」の目次

はじめに
1． STAMP解説
2． STPAの手順（全体説明）
3． 対象システム概要
4． STPA分析実施例の説明
5． Advanced Technic
6． エンタープライズ系システムでのSTAMP適用
7． まとめ
おわりに

第１節と第２節には、我が国におけるSTAMP/STPAのエキスパートによる解説をつけた。STAMP/STPAの初学者にとって分かりやすい入門解説として有用であろう。当WGで分析対象とした鉄道の踏切制御システムについて第３節で述べ、分析の結果を第４節に示した。第５節には、支援ツールの活用、並びに、従来のシステム記述や分析によるシステムの振る舞いや特性の理解に基づいてSTPAを適用する試みについて述べた。第６節では、エンタープライズ系のシステムに対するSTAMP適用についての検討結果を示した。

4.2 手順解説書の特徴

以下に、本小冊子の特徴を記す。

分析作業の各Stepで、【Input/Process/Output】を整理して説明している。

表4 Input/Process/Outputの例

作業名称	UCA（Unsafe Control Action：非安全制御動作）の抽出
目的	ハザードにつながり得る制御動作の不具合を識別する（発想する）
入力	①UCAを導き出すための４つのガイドワード（４分類） ②アクシデント、ハザード、安全制約の一覧表 ③制御構造図
処理	①UCA識別の表を準備する ②最上列に４つのガイドワードを記す ③最左行に制御構造図中にある制御をすべて記す ④各マスごとに、当該（最左行の）制御動作が当該（最上列）状況になった場合、いずれかの安全制約違反に成り得るかを考える。 ⑤安全制約違反に成り得るならば、UCAであると判断する
出力	①縦軸：制御行動、横軸：ガイドワードとしたUCA一覧表。
	想定外を排除することを忘れないように。

実施例では、分析結果だけではなく、分析の際に作成した中間情報も含めて記載した。

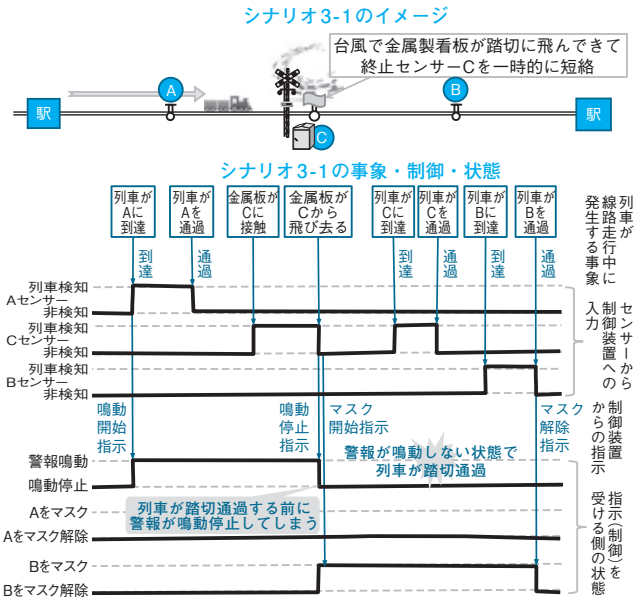

図６ 分析時に作成した中間情報の例

当WGが実際に具体的なシステムのSTPAによる分析を行い、分析結果について、STAMPの専門家、鉄道分野の専門家と議論して得た知見を整理し、勘所として解説している。

【勘所】
登場人物を整理する際には、要求仕様書に記載された登場人物のみをリストアップして、実装依存の登場人物が現れないようにする。
：
●制御とデータは区別しやすいようにする。制御は青、データは赤のように色分けしたり、制御は下向き矢印、データは上向き矢印に統一したり、など
：
コントロールストラクチャーにおけるブロック（コンポーネント）の数は4つ程度にするのが良いと言われている。

5 おわりに

IPA/SECが発行した「はじめてのSTAMP/STPA」が、システムの安全性に関するモデル化及び分析方法に関して、いくばくなりとも有用な情報を提供するものとなれば幸いである。

2015年度は主に、単線踏切制御という比較的にシンプルなシステムを取り上げて安全分析の検証を行ってきた。今後は、コンピューターシステムに人・組織が絡む複雑なシステムや、人と組織、組織と組織のプロセスに関する安全分析へのSTAMP適用可能性についても検証を行い、そこから得られた知見を公開していきたい。

2016年12月５、６日に九州大学にて、日本ではじめてのSTAMPワークショップが計画されているが、多くの産業界からの参加を期待したい。

URL：https://sites.google.com/site/stampwsj/

【脚注】
※3 http://www.ipa.go.jp/sec/reports/20160428.html