基板検査システム開発におけるシステム安全性解析手法（STAMP/STPA）導入の試み

吉田 邦雄

従来、基板検査システムのシステム設計ではアクティビティ図やシーケンス図を作成してシステムの振る舞いを表現してきた。しかしながら、これらで表現できるのは一般的な振る舞いに留まっているのが現状であり、システムが高機能化・複雑化する中で設計段階において異常系の振る舞いを精緻に特定して不具合の混入を防ぐことは難しい。

筆者らは基板検査システムの開発においてシステム安全性解析手法の 1 つである STAMP/STPA の適用を試みた。その結果、従来からあるシステム設計手法では抽出しづらかったであろう不具合発生要因を設計初期段階にて導出することができ、今後、更にシステムが高機能化・複雑化したとしても、信頼性を保ち続けることができる手応えを得た。

本稿では、基板検査システムの開発に STAMP/STPA に適用した分析事例を述べると共に、本取り組みを通じて得られた分析作業の効率化手段を提案する。

Introduction of System Safety Analysis Method (STAMP/STPA) in the PCB Inspection System Development

YOSHIDA Kunio

Traditionally, system behavior has been expressed by creating activity and sequence diagrams in the system design of the PCB Inspection System. However, it is difficult to specify the abnormal behavior of the systems in the design phase to prevent the inclusion of defects, as the system becomes highly functional and complicated.

The authors have tried to apply STAMP/STPA which is one of the system safety analysis methods in the development of the PCB Inspection System. As a result, it has been able to acquire the cause of failure that would have been difficult to extract with existing system design methods, and it helps to maintain high reliability of the system even if it becomes more sophisticated and complex in the future.

This paper describes an analysis case applied to STAMP/STPA for the development of the PCB Inspection System and proposes a method for improving the efficiency of analysis work obtained through this approach.

1. まえがき

近年、自動運転技術の発展を支える ADAS（Advanced Driver-Assistance Systems、先進運転支援システム）や、5G を代表とする通信技術の発展に伴い、SMT（Surface mount technology、基板実装技術）は高密度化、微細部品化が進んでいる。また、社会インフラを支える電子機器の製品安全を実現するために SMT には高い信頼性と安定性が求められている。SMT の最終工程において、部品が正しく実装されているか、基板と部品が正しくハンダ付けされているか検査を行う基板検査システムは重要な役目を担っている。

基板検査システムに求められる機能も近年、多機能化かつ高機能化している。多様な機能を実現するために、多くのシステムホスト間、アプリケーション間のコミュニケーションが発生することで、システム構成は複雑化の一途を辿っている。加えて、ビックデータや AI 向けに画像データの重要性は高まっており、工場内ネットワークにおいて大量・大容量の画像データが高頻度で送受信や読み書きが行われる環境下でも、安定動作し続ける高い信頼性が求められる。

従来、システム設計ではアクティビティ図やシーケンス

Contact : *YOSHIDA Kunio* kunio.yoshida@omron.com

図を作成してシステムの振る舞いを表現してきた。しかしながら、これらで表現するのは一般的な正常系と、代表的な異常系の振る舞いに留まっているのが現状であり、設計段階で異常系の振る舞いを精緻に特定して不具合の混入を防ぐことは難しく、社内のテスト環境では再現できない問題が、特定の顧客環境で発覚するような不具合が発生していた。

「ZERO DEFECT ～不良を作らない～」生産ラインの構築を目指すオムロンの基板検査システムに求められる高機能性と高信頼性の両方を両立するための設計分析手法として、システム思考に基づく新しい安全性解析手法 STAMP/STPA の適用を試みた。STAMP/STPA は複雑なシステムにおいても以前は運用でしか発見できなかった「想定外の想定外（Unknown unknowns）」が、開発プロセスの初期に識別され、除去または低減することができるとされている[1)]。

本稿では基板検査システムの開発に STAMP/STPA に適用した分析事例を述べると共に、本取り組みを通じて得られた分析作業の効率化手段を提案する。

2. STAMP/STPA

2.1 STAMP/STPA とは

STAMP/STPA（Systems-Theoretic Accident Mode and Processes/Systems-Theoretic Process Analysis）はマサチューセッツ工科大学の Nancy G Leveson 教授が提唱している、システムの構成要素間の相互作用によって発生する問題を分析する手法である[1,2)]。航空宇宙分野での適用[3)] のみならず、社会インフラ領域への適用が広まっている[4)]。

従来から存在する分析手法 FTA（Fault Tree Analysis）や FMEA（Failure Mode and Effect Analysis）は機器や組織の単一故障を分析する手法として 1960 年代から存在している。これらの分析手法は現代において進化し続ける複雑なシステムにおいては限界がある。なぜならば、複雑なシステムにおいては単一コンポーネントの故障のみならず、コンポーネント間のコミュニケーション・ミスマッチが事故を引き起こすと考えられるためである。Nancy G Leveson 教授が提唱している STAMP/STPA はトップダウンプロセスをとり、システム内の各コンポーネントが直接的、間接的に発生する相互作用を俯瞰的に捉えて偶発的に発生する創発特性を制御して事故を防止する考え方に基づく。

2.2 STAMP/STPA 分析手順

STAMP/STPA 分析の手順を以下に示す。

Step0：（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、ハザードを定義する。アクシデントは利害関係者にとって受け入れられない何らかの価値が喪失するという広い意味で定義される。ハザードとはアクシデント一歩手前の、放置してはいけない状態である。最後に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。

Step0：（準備 2）コントロールストラクチャの構築

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を分析し、コントロールストラクチャを構築する。図 1 に一般的なコントールストラクチャを示す。

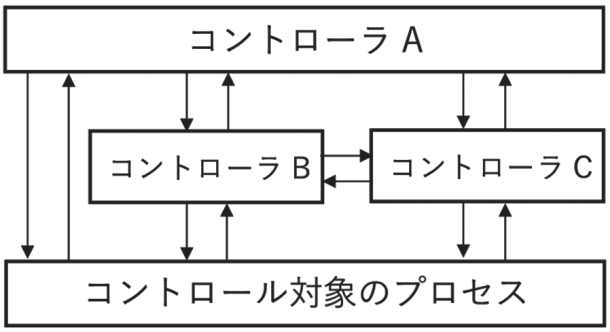

図 1　一般的なコントールストラクチャ

Step1：非安全なコントロールアクション（UCA）の抽出

安全制約の実行に必要なコントローラによるコンポーネント間の指示（Control Action）を識別して、識別された指示の中から UCA（Unsafe Control Action、非安全制御動作）を抽出する。UCA を抽出する際のヒントとして 4 つのガイドワードがある。

- Not Providing：与えられないとハザード
  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。
- Providing causes hazard：与えられるとハザード
  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。
- Too early/too late, wrong order causes hazard：早過ぎ、遅すぎ、誤順序でハザード
  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、早すぎて、遅すぎて、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。
- Stopping too soon/applying too long causes hazard：早過ぎる停止、長過ぎる適用でハザード
  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、もしくは適用が長過ぎることがハザードにつながる。

Step2：ハザード要因（HCF）の特定

Step1 で抽出した UCA 毎に、関係するコントローラと制御対象プロセスを識別してコントロールループ図を作成し、

因果関係シナリオ生成の抽象モデルを参照しながら HCF（Hazard Causal Factor、ハザード誘発要因）を特定する。

コントロールループ図とは、着目した UCA に関係するコンポーネントのみを抜き出した図であり、複数のコンポーネント間の相互作用を同時に考えるではなく、二つのコンポーネントに絞って考えるためのものである。

因果関係シナリオ生成の抽象モデルとは、HCF の特定を支援するモデルであり、HCF の一般的発生理由が記載されている。

3. 基板検査システム開発への導入

3.1 事前準備

本取り組みは図 2 に示す検査システム事業部の開発プロセスのうち、赤い囲み線で示した概念設計からシステム設計フェーズの間にかけて実施した。

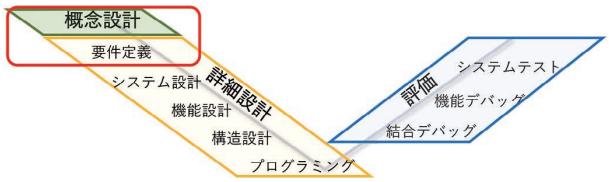

図 2　開発プロセスと STAMP/STPA 分析の実施タイミング

分析に際し、事前準備として図 3 に示すシステム全体の概略図を作成し、システムに対する理解を分析者の間で共有できるようにした。システム構成要素の役割について説明する。

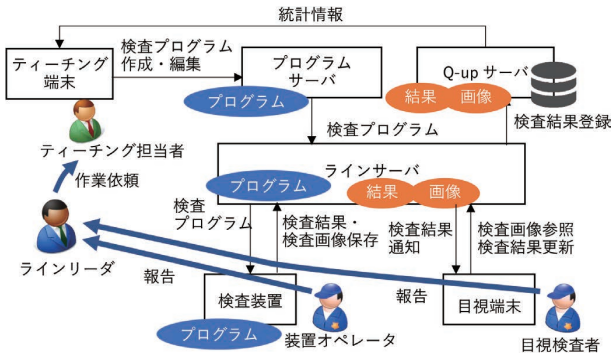

図 3　システム概略図

・ティーチング端末：基板検査に必要な検査プログラムの作成、編集を行う。
・プログラムサーバ：検査プログラムを保存する。
・ラインサーバ：いくつかの生産ライン毎に設置し、検査プログラムのレプリカサーバとしての役割と、検査結果および検査結果画像の一次保存場所となる。
・Q-up サーバ[5]：Database System を搭載して検査結果の分析を実施する。
・検査装置：基板の良否判定を実施する。
・目視端末：検査装置が不良品判定した基板に対し、目視検査作業者の目視検査によって良否の最終判定を実施する。

3.2 Step0（準備 1）：アクシデント、ハザード、安全制約の識別

システムのアクシデント、ハザード、安全制約は以下のように定義した。

アクシデント：
・オムロンが提供する検査システムアプリケーションが原因で生産ラインが停止する

ハザード：
・検査装置の起動に時間がかかり検査を開始できない状態
・検査機で検査プログラムの読込みができず、検査機種切り替えができない状態
・目視端末で、目視検査対象の基板が端末内に到着しているのにも関わらず目視検査が実施できない状態
・検査プログラムの内容に不整合が生じ、検査機が異常を検知して停止する状態
・基板が装置手前に到着して装置内に基板が無いのにも関わらず、基板を搬入しない状態
・検査が完了して下流装置が搬入可能状態にも関わらず、基板を搬出しない状態

安全制約（ハザードの裏返し）：
・生産開始のタイミングまでに検査が実施できる状態でなければならない
・検査機で検査プログラムが正常に読み込めて、段取りが実施できなければならない
・目視端末に基板が到着した時点で目視検査を開始できなければならない
・検査プログラムの不整合によって検査機が異常停止してはならない
・基板が到着し、装置内に基板が無いのであれば基板を搬入して検査を開始しなければならない
・検査を完了して下流装置が搬入可能状態ならば速やかに基板を搬出しなければならない

3.3 Step0（準備 2）：コントロールストラクチャの構築

本ステップで構築したコントロールストラクチャを図 4 に示す。この図が示すように、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のが、STAMP/STPA の特徴でもある。

図 3 に示した概略図とはシステム構成要素に変化がある。これは、設計の過程において変更が発生したことを表している。

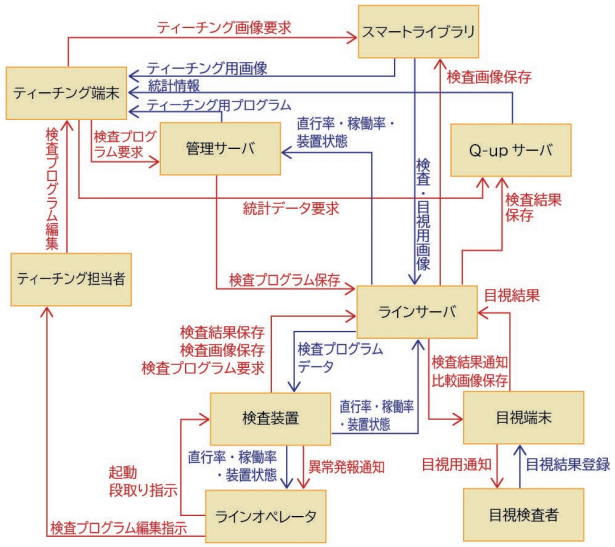

図 4　コントロールストラクチャ

3.4 Step1：非安全なコントロールアクション（UCA）の抽出

総計 51 個の UCA を抽出した。抽出した UCA の例を示す（表 1）。

表 1　抽出した UCA の例

CA	検査プログラム要求
From	検査装置
To	ラインサーバ
Not Providing	・検査プログラムの取得が必要にも関わらず段取りに必要な検査プログラムを要求しない。
Providing causes hazard	・間違った検査プログラムを要求する。 ・検査プログラムの保存に失敗する。
Too early/too late, Wrong order causes hazard	・ラインサーバの負荷が高くなり、他の検査機に対する処理が実施できなくなる。 ・検査プログラムの要求開始が遅れ、生産開始時に検査プログラムが利用できない。
Stopping too soon/applying too long causes hazard	・検査プログラムの保存に時間が掛かり過ぎて生産時に検査プログラムが利用できない。

UCA の抽出においては、文脈にコンテキスト（状況）を含めることが重要である。コンテキストを含んだ UCA の表現がしにくい場面に遭遇した際には「～にも関わらず、～だ」という表現を用いるよう心掛けた。例えば「検査結果の出力が遅れる」という UCA があった場合に、「次の検査対象基板到着するまでに検査結果の出力が完了しなければならないのにも関わらず、検査結果の出力が完了しない」という表現にする。

3.5 Step2：ハザード要因（HCF）の特定

本取り組みでは総計 85 個の HCF を抽出した。抽出した HCF の例を示す（表 2）。

表 2　抽出した HCF の例

HCF	HCF シナリオ
管理ファイルと実体ファイルの乖離	新しいプログラムがサーバ上にあるのにも関わらず、検査装置が新しいレビジョンのデータを認識していない。
管理サーバの排他処理誤り	複数のティーチング端末で同時に検査プログラムの更新が行われたときに、検査プログラムまたは検査プログラムの管理情報が正しく保存されない。
ラインサーバの負荷	ラインサーバの負荷が高くなり、検査装置が検査プログラムを要求してもレスポンスが無い、または遅れる。
不正なリトライ処理	検査プログラムの保存に失敗したので繰り返し要求処理を行う。

問題を単純に扱えるように HCF の抽出において、以下の点に留意した。

風が吹けば桶屋が儲かるということわざがある。一見すると全く関係がないと思われる場所・物事に影響が及ぶことの喩えであり、因果関係を深く追っていくと、何もかもが問題の要因となり得る事を示唆している。本取り組みでは問題をシンプルに取り扱うために、因果関係は 1 次までにとどめるようにした。風が吹くと桶屋が儲かるの例で例えると、突風で砂ぼこりが立つ、その結果、砂ぼこりが目に入り視力を失う人が増える、までに留める。HCF の表現にならうと「突風により砂ぼこりが立つ」とし、その対策として「風が強い日は水を撒いて砂ぼこりが舞わないようにする」といった具合である。

図 4 で示したように、基板検査システムは複数のコンポーネントがネットワークで接続された構成をしている。STAMP/STPA の手順ではコンポーネント間の指示に着目し、コンポーネント間の相互作用によって発生する問題を抽出するため、同じような HCF が複数のホスト間で重複して抽出された。対策を検討するにあたり HCF 個々に対策を立てるのではなく、HCF を集約し、集約された HCF ごとに対策を検討するようにした。結果として HCF のばらつきや抽出の漏れを炙り出すとともに、対策検討をしやすくする効果があった。集約した HCF と HCF 抽出シナリオの例を示す（表 3）。

表 3　集約した HCF の例

集約 HCF	HCF シナリオ
サーバ負荷が高くなりデータを受領しない／返却しない	・複数のティーチング端末が同時に検査プログラムの更新作業を開始して、検査プログラムの保存の要求が同時多発的に発生する。 ・検査プログラムの連続保存操作によってディスク負荷が高くなる。 ・検査プログラムの保存に失敗したので繰り返し要求処理を行う。 ・ラインサーバにつながる装置で同時多発的に大きなサイズの検査画像の保存が行われた。
管理情報と実態の間に矛盾が発生する	・新しいプログラムがサーバ上にあるのにも関わらず、管理情報の更新直前に検査装置が古いレビジョンのデータを要求する。 ・検査プログラムの実体ファイルの保存が未完のうち、管理ファイルのみ更新され、そのタイミングで検査プログラムを利用しようとして、矛盾が発生する（またはその逆）。 ・管理情報上では存在するはずの画像がストレージ上に存在しない。
破損した・データ整合性の取れていないファイルの保存、データ	・ファイル保存中にキャンセルを実行して、ごみデータが残る。 ・保存処理中に例外が発生して、中途半端なデータが残る。 ・複数のトランザクションからなる処理で、最後のステップでネットワーク切断が発生した結果、冒頭に実施したトランザクションの結果をクリアしていない。

3.6 HCF 対策の立案

STAMP/STPA で定義された手順は HCF の特定までであるが、基板検査システム開発プロセスの中で抽出された各 HCF を排除するための方策を検討し、システム仕様あるいは、各アプリケーションの機能仕様にフィードバックする。

4. 考察

STAMP/STPA のプロセスを実施することで新たに抽出できた UCA としては以下のような内容があった（表 4）。

表 4　STAMP/STPA のプロセスにて抽出できた UCA 例

CA	検査プログラム保存
From	管理サーバ
To	ラインサーバ
Too early/too late, wrong order causes hazard	連続的に検査プログラムの更新作業を実施して検査機にプログラム本体が行き渡る前に管理データが更新されて、データの整合が取れなくなる。

CA	検査結果保存
From	ラインサーバ
To	Q-up サーバ
Stopping too soon/applying too long causes hazard	Q-up サーバへの検査結果保存処理に時間がかかって、ラインサーバが待ち状態となり、その間検査機からの検査結果を保存できない。

コントロールストラクチャをベースに検討することで、システム全体を俯瞰でき、本来の目的を意識したうえで、構造上不足している点を見つけることができた。

一見するとよくある不具合のように見受けられるが、これらアクティビティ図やシーケンス図では振る舞いを表現しづらく、処理タイミングによっては問題が発生する場合があるようなリスクを抽出できている。これらのリスクに対し、開発終盤のテスト段階もしくは顧客環境にて実際に問題が発覚してから個別に対処するのではなく、開発プロセスの初期から問題を把握し、分析を通して得られた図表を介して開発メンバー間で共有した上で本質的な対応を取れることは、開発プロジェクトに QCD の観点で貢献するものと考える。

STAMP/STPA の導入における分析作業は、Step1：UCA の抽出件数ごとに、Step2：HCF 特定を繰り返すため、分析作業量が多くなる。本手法の活用を広めるためには STAMP/STPA が持つ創発特性の制御能力を維持しつつ、分析作業の効率化が必要であると考える。本取り組みでは、HCF 対策検討時に個別に抽出した HCF の集約を行い検討立案の効率化を図った。更なる効率化の手段として、Step1：UCA の抽出までを実施した後で、同じようなコンテキストで発生する UCA の集約を行い、集約された UCA に対して HCF を検討する方法が考えられる。特に、基板検査システムのような分散システムにおいては、ホスト間のネットワーク接続に起因する HCF は集約して検討できるため有効である。

5. むすび

今後、求められる要求の多機能化・高機能化が予測される中でも高品質なシステム開発の実現に向けて、基板検査

システムの設計に STAMP/STPA を適用し、オムロンが提供する検査システムアプリケーションが原因で生産ラインが停止することをアクシデントと定義して STAMP/STPA による安全性解析を試みた。

システム全体をコントロールストラクチャで表現して俯瞰しながら、ガイドワードを照らし合わせてコンポーネント間の指示によって発生し得る問題を抽出することで、アクティビティ図やシーケンス図の作成だけでは抽出できないような異常系の振る舞いに対するリスク項目を導き出すことができた。

STAMP/STPA 分析の過程でシステム全体を可視化して問題とその問題に至るまでのコンテキストおよび、対策のトレーサビリティを残せることでドメイン知識の形式化に寄与すると共に、今後、更にシステムが高機能化・複雑化したとしても信頼性を保ち続けることができる手応えを得た。

一方で、分析作業には根気と労力を要する。開発プロジェクトにおける STAMP/STPA の適用障壁を下げて、活用機会を増やすためには、分析作業の効率化が不可欠だと考える。本取り組みでは、個別に抽出した HCF を集約し、対策検討の効率化を図った。また、同じようなコンテキストを有する UCA の集約による、HCF 抽出の効率化を提案した。

今後もこの STAMP/STPA を活用し、システム製品の信頼性向上に貢献していくと共に、分析時に工夫した点や基板検査システムならではの肝所を蓄積し、より設計に適用しやすくなるよう努めていく所存である。

参考文献

1）Leveson, Nancy G.; Thomas, John P. “STPA Handbook”. http://psas.scripts.mit.edu/home/get_file2.php?name=STPA_handbook_japanese.pdf,（参照 2020-8-4）.

2）情報処理推進機構.“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～”. 情報処理推進機構. https://www.ipa.go.jp/files/000051829.pdf,（参照 2020-8-4）

3）氏家 亮. JAXA コウノトリプロジェクトへの STAMP/STPA 適用例. 計測と制御. 2016, Vol. 55, No. 5, p. 405-409.

4）北村 知. JR 東日本における STAMP 活用の取り組み. SEC journal. 2018, Vol. 13, No. 4, p. 30-37.

5）オムロン株式会社.“「品質起点」の生産性向上支援ソフトウェア Q-up System”. https://www.fa.omron.co.jp/data_pdf/cat/q-upnavi_scwb-046a-1.pdf?id=1463,（参照 2020-10-20）.

執筆者紹介

吉田 邦雄　YOSHIDA Kunio

インダストリアルオートメーションビジネスカンパニー
検査システム事業部 開発部
専門：ソフトウェア工学