STAMP初心者を卒業する

有人宇宙システム株式会社　野本 秀樹

1 はじめに

ナンシー・レブソンによる“A STPA Primer”[1]は、STAMPの入門書として非常に優れたものである。ここに記された様々な事例や深い洞察の助けを借りてSTAMPのアナリストは多くの安全分析を行ってきた。

しかし、“Primer”は、文字通り「初心者のための手引き」である。プロフェッショナルなSTAMPアナリストになるためには、ここを卒業し、その先に向かって道を切り開いていかねばならない。

本稿では、STPA Primerからどちらの方向へ向かうべきかを、Primerを参照しながら論じてみたい。

2 STAMPの神話

STAMPを始めたばかりのアナリストが陥りやすい間違いが、このSTPA Primerに書かれている。

“FAQ: Does the control structure always have to be hierarchically linear, like a ladder?
No. The control structures in this primer are only examples from various applications, and yours may be different. Although some control structures look like a ladder, like the generic operations and development structure shown in Figure 1.5 or the simple Aircraft -> Pilots -> ATC structure for the NextGen ITP example in Figure 2.2, this is not always the case. For example, the high-level control structure for the Gantry 2 radiation therapy machine shown in Figure 2.9 shows how control and feedback paths can skip across levels in the control structure.” (“Primer” p.41)

よくある質問：制御構造図は常にきれいな階層構造である必要がありますか？
答え：いいえ。

STPA Primerをはじめ、多くの論文に現れる制御構造図は制御構造が階層化されており、常にControlled Processは単一のControllerを持っている。しかし、現実は必ずしもその通りではない。Primerにも書かれているように、安全を制御している構造というものは、もっとバラエティに富んでおり、レブソンの言う創発的な問題を生む可能性のあるシステムにこそ、通常とは異なる複雑な制御関係が存在していると考えるべきである。本稿では、Primerに一個所だけ書かれている制御構造図のないセクションに書かれた事例を対象に分析を行う。この事故事例について、詳細の制御構造図があえて書かれておらず、読者の今後の課題のように紹介されているのは、この事例が正に単純ではない制御構造の事例であり、初心者であることを卒業するための、読者への「宿題」だからである。

3 Multiple Controllers

“Primer” p.20にある事例には唯一制御構造図が書かれていない。そこで、Primer卒業のため、ここで、どんな制御構造図が書けるのか試してみる。

事故の概要は、以下のようなものである[2]。

2002年、ドイツユーバーリンゲンで起こった航空機の衝突事故である。2機（2937便と611便）はそれぞれ2937便が真西に向かって、611便は真北に向かって飛行中であった。

それぞれには2種類の安全コントローラが存在している。

TCASとATCである。そのうちの一つTCAS（Traffic Collision Avoidance System）は、他機の接近を検知して、パイロットに対

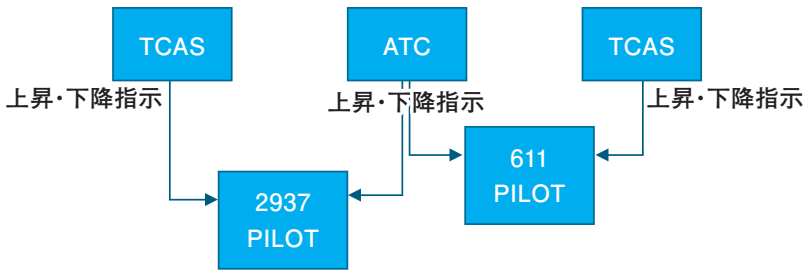

図1　TCAS/ATC/PILOT間の制御構造

して警告を発し、上昇・降下・方向・速度調整を指示する自動システムである。

もう一つのATC（Air Traffic Control）は、航空機の運航を取り仕切る管制業務の一般名称であり、こちらは、航空管制官が状況によってパイロットに指示を出す。

2つの安全システムは互いに独立であり、どちらかがどちらかに従属していないため、システムエラーや管制官のミスなどに互いに影響されない。つまり、完全な独立二重系による安全性の向上を目指した制御構造である。

2機は、それぞれがTCASとATCによって安全を制御される図1のような関係にあった。

まず、2機に搭載されたTCASが双方の機影を検知した。7秒後に地上のATC管制官が危険を察知し、それぞれに「2397は上昇」「611は下降」の指示を出した。

一方、機体搭載のTCASは、2397に対しては「下降」、611に対しては「上昇」を指示していた。

結果、2397は管制官の指示に従い「上昇」、611もTCASの指示に従い「上昇」し、両機が空中で衝突した。

この事故を分析するとき、制御構造図に書かれた「上昇指示」や「下降指示」を個別に分析しても、得るものは少ない。「上昇指示が来ない場合」も「上昇指示が遅れる場合」も、「間違った上昇指示が来る場合」も、すべてハザードの発生要因である。しかし、ここでの最大の問題は、そのような個々の問題ではない。2つのcontrolled processに対する2つのcontrol actionが、2つのcontrollerから同時に発せられるときの問題である。もし、2つのcontrollerからの指示が逆になってしまうと、2機は衝突してしまうという問題に言い換えることができる。

正常状態の組み合わせは表1の4通り、更に、TCAS故障時の組み合わせは表2の通り8通りである：

表1　正常状態におけるATC指示値とTCAS指示値の影響

	ATC指示値		TCAS指示値		衝突?
	2397向け	611向け	2397向け	611向け	
(1)	↑	↓	↑	↓	
(2)	↑	↓	↓	↑	衝突
(3)	↓	↑	↑	↓	衝突
(4)	↓	↑	↓	↑	

表2　TCAS不調時におけるATC指示値とTCAS指示値の影響

	ATC指示値		TCAS指示値		衝突?
	2397向け	611向け	2397向け	611向け	
(5)	↑	↓	—	↓	
(6)	↑	↓	—	↑	衝突
(7)	↓	↑	—	↓	衝突
(8)	↓	↑	--	↑	
(9)	↑	↓	↑	—	
(10)	↑	↓	↓	—	衝突
(11)	↓	↑	↑	—	衝突
(12)	↓	↑	↓	—	

表1、2に示したように、2つのcontrollerによる指示を出すシステムは、正しい指示が行われている場合に限定したとしても、確率50％でしか保証されていない（表1）。もともとTCAS不能時のために維持されているATCであるが、TCAS故障状態を安全化できているとは言えない。確率は相変わらず50％である（表2）。もし確率50％であれば、この安全装置があることで得られる安全上の利得はゼロということになってしまう。どのようにTCASの信頼性を上げても、この確率は改善できない。表1＋表2の50％が表1単独の50％に戻るだけである。

現在では、この事故をきっかけにルールが明確になり、「TCASと地上管制官の指示に食い違いがある場合はTCASに従う」ことになっている。すなわち、制御構造は以前のままの3頭立てのcontrollerであるが、PILOTのプロセスモデルに、「両者が食い違った場合は、TCASを優先する」というロジックが追加された。

しかし、実は、肝心のTCAS故障時に、このルールは安全性を高めない。

TCASが故障している場合は、表2のように、パイロットは地上管制に従わざるを得ないが、それでは、この事故が全く同じ理由で再発してしまうからである。現在のルールのように、TCASの信頼性を多とし、地上管制をヒューマンエラーの発生源のごとく扱うルールでは、上記の表1の問題が解決されるにすぎないのである。

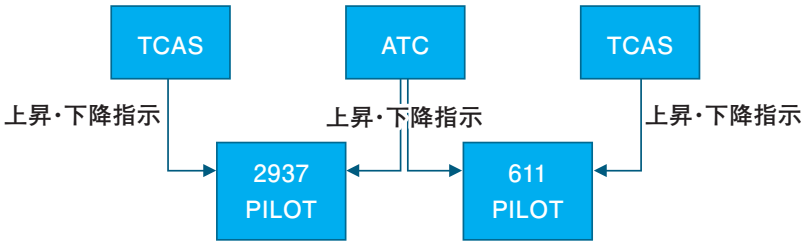

図2　ルール制定後の制御構造は同じ
パイロットのプロセスモデルが複雑化した

図1の制御構造が変更を受けないまま、図2のように、人間の持つべきプロセスモデルが複雑化される例は多い。ヒューマンエラー防止のための自動化が行われているが、完全自動化ではなく、システム故障時には人間に責任が帰ってくる自動運転などがその典型例である。通常のドライバーであれば難なく安全化可能と思われるシチュエーションでも、自動運転モードが突然手動運転モードへ遷移すると、ドライバーの多くは安全な回避行動を取ることができなくなることが、自動車技術会（JARI）の実験で明らかになっている[3]。

4　問題の解決

このような問題点を解決するためには何が必要になるであろうか。

STAMPが提示した新しさこそが、この問題を解決する決め手となる。STAMPの神髄は「安全制御構造」のモデル化である。従って、我々は安全制御構造そのものを見直すことによって安全化を目指す。図2のような独立した3頭立て馬車による制御ではなく、図3のように2つの情報源であるTCASとATCが協調関係を築く制御構造が必要となる（“Primer” p.20 Figure 1.9）。

しかし、この「調停」による協調をTCAS側に担わせることは困難である。そもそもTCASが故障した場合には、そのような仕組みは役に立たない。

従って、これは人間が担うべき役割となる。管制官は2機のTCASの指示を入手する、若しくは、2機のTCASと全く同一の答えを出すTCASシミュレータを地上に持ち、その指示と同じ指示をパイロットに指示するという仕組みが必要となる。図3の青太矢印の「調停」がこのために追加となる。これは、制御アクションの追加という小さな変更ではなく、TCAS中心の制御構造からController間の協調による制御構造への変革を意味する。この制御構造の変更によって初めて、TCAS不能時にはATCが、ATC不能時にはTCASが安全化を担保できる真の二重系となり、正常時にも異常時にも高い信頼性を発揮できるようになるのである。ちなみに、このような協調安全はSafety2.0と呼ばれている[4]。Safety 2.0の目指す安全は、これまでのような「ヒューマンエラー防止のための自動化」という、図2のようなものではなく、図3で実現される「人と機械の協調的安全」である。機械が人のエラーを監視するという対立構造ではなく、機械と人が互いに助け合う関係が、新たな時代の安全化の主流となると期待されている。

5　まとめ

STAMPの初心者を卒業し、プロフェッショナルなアナリストになるためには、制御構造の分析眼と真に安全な制御構造を設計できる能力、すなわちシステムズエンジニアリング能力が必要になる。本論で示したように、現在安全と信じられている仕組みも、制御構造を分析してみれば、実は問題を抱えている可能性がある。事故後に修正された航空管制のルールはTCASという自動警報システムを管制官の判断に優先させよ、というものであるが、この修正では真の安全は達成できない。真の安全を達成するためには、3頭立て馬車のいずれかを優先させるのではなく、3頭の馬の間の協調を「構造の中に創り込む」ことが必要である。これこそ、レブソンがかねてより主張している「システム理論に基づく安全化」であり、新しい安全、Safety 2.0への道である。

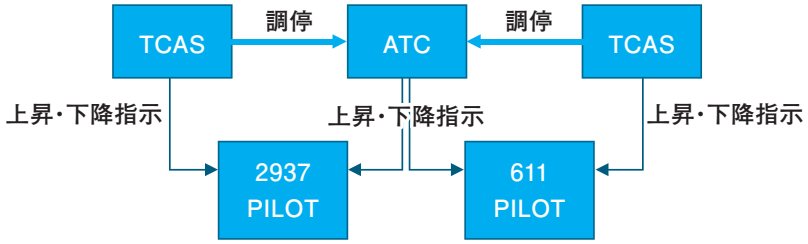

図3　新たな制御構造

【参考文献】

[1] Leveson (2013), “A STPA Primer v.0” http://sunnyday.mit.edu/STPA-Primer-v0.pdf

[2] Gallagher Paul (2002) “Jet pilot’s 14 seconds dilemma before fatal crash”, The Scotsman (Edinburgh, UK) (http://news.scotsman.com/germanplanecrash/Jet-pilots-14-seconds-dilemma.2341758.jp)

[3] 本間, 若林, 小高, (2017)「高度自動運転における権限移譲方法の基礎的検討（第4報）」公益社団法人自動車技術会2017年秋季大会学術講演会

[4] 中村英夫（2017）IoT時代の新しい安全「Safety 2.0」の全貌 ET2017 独立行政法人 情報処理推進機構 SEC先端技術入門ゼミ