STAMPによる
複数コントローラ間の協調安全解析

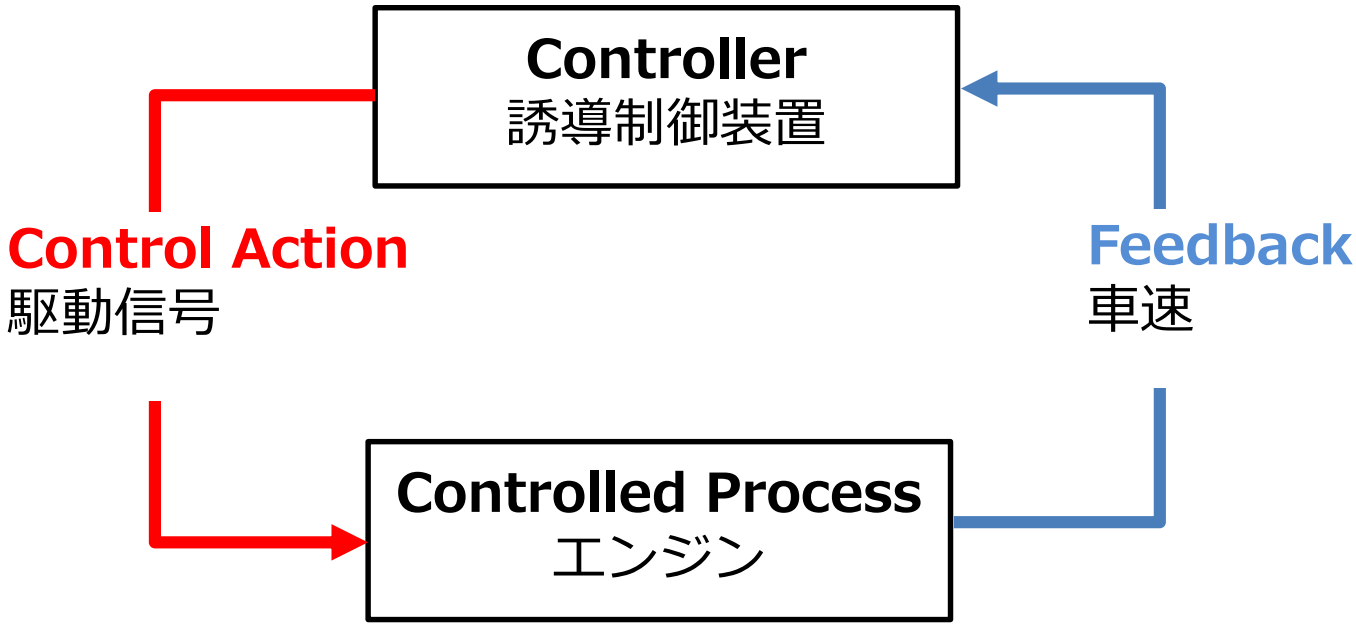
野本秀樹
有人宇宙ステム株式会社
IV&V研究センター長
nomoto.hideki@jamss.co.jp

STAMPの手法

- システムの構造を「安全制御構造」(Safety Control Structure）と名づけ、これを作図するところからはじめる。
  - 事故の原因、ハザードの原因をいきなり考えるのではなく、まず、システムの「構造」を明らかにすることからはじめる
  - 安全解析も不具合解析も全て同じ
  - システムの構造=「安全制御」の仕組み

システムの制御構造

- Control Structure Diagram
  - 以下のようなフィードバックループを基本形とする
  - STAMPは、「制御する」Controllerによって安全が達成され、
  - その「制御」が破綻することによって事故が発生すると考える



ハザード発生要因の分析方法

ハザード：急加速

Control Action	提供されないとハザード	提供されるとハザード	開始タイミングが悪いとハザード	終了タイミングが悪いとハザード
アクセル駆動信号	特定のモードで、アクセル駆動信号をマスクすると、ドライバーが強くアクセルを踏むように誘導してしまう。	アクセル開度に従っていればハザードにはならない	駆動が遅すぎると、ドライバーが強くアクセルを踏むように誘導してしまう。	アクセルを踏んでいるのに加速がとまってしまうと、ドライバーが強くアクセルを踏むように誘導してしまう。

制御構造が単純なら、このように、個別の制御アクションを分析しても良い。

ハザード発生要因の分析方法

ハザード：急加速

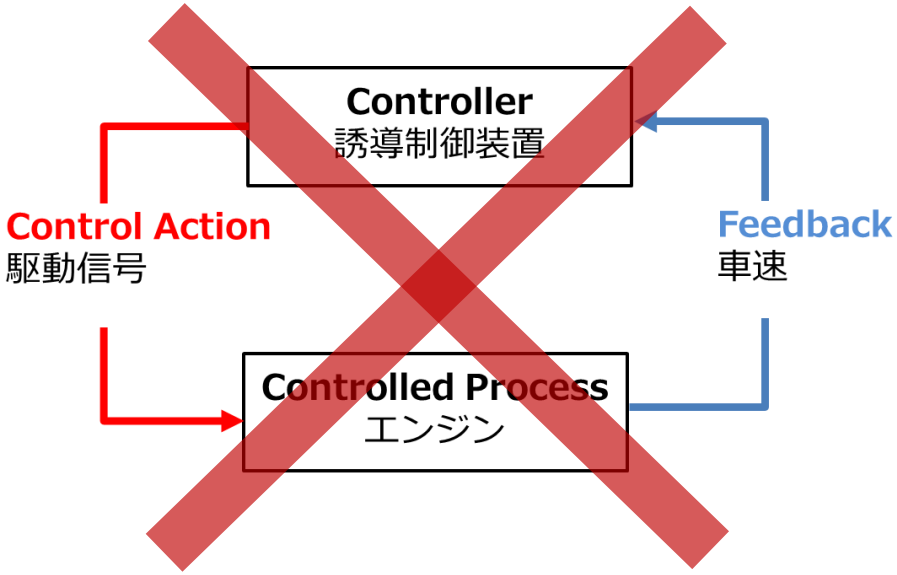
Control Action	提供されないとハザード	提供されるとハザード	開始タイミングが悪いとハザード	終了タイミングが悪いとハザード
アクセル駆動信号	特定のモードで、アクセル駆動信号をマスクすると、ドライバーが強くアクセルを踏むように誘導してしまう。	アクセル開度に従っていればハザードにはならない	駆動が遅すぎると、ドライバーが強くアクセルを踏むように誘導してしまう。	アクセルを踏んでいるのに加速がとまってしまうと、ドライバーが強くアクセルを踏むように誘導してしまう。

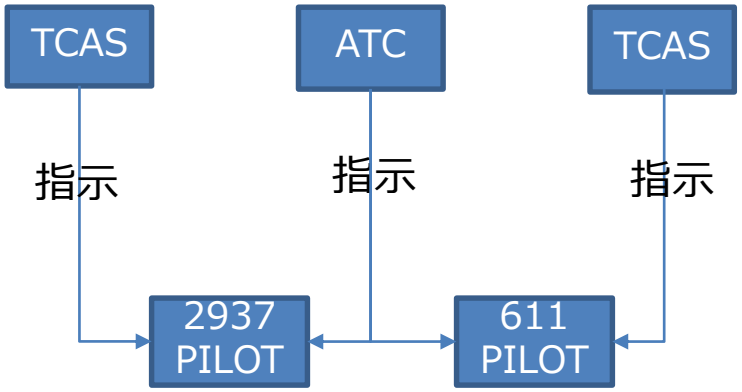
日本では、このガイドワードを使った手順がSTAMPの手法と考える人が多い。

しかし、これでは、HAZOP手法と基本的に同じ。

複数コントローラ

世の中のシステムの多くは、このような単純な構造ではない

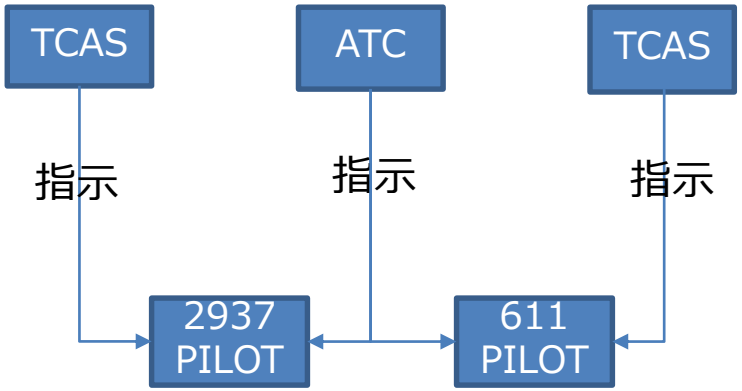




複数コントローラ

しかし、世の中のシステムの多くは、このような単純な構造ではない

単純ではない構造を持つシステムこそが、STAMPの本領を発揮する場。

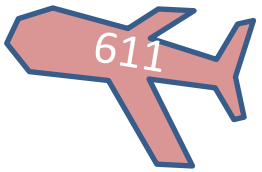


複数コントローラ

2002年ドイツでの
航空機衝突事故

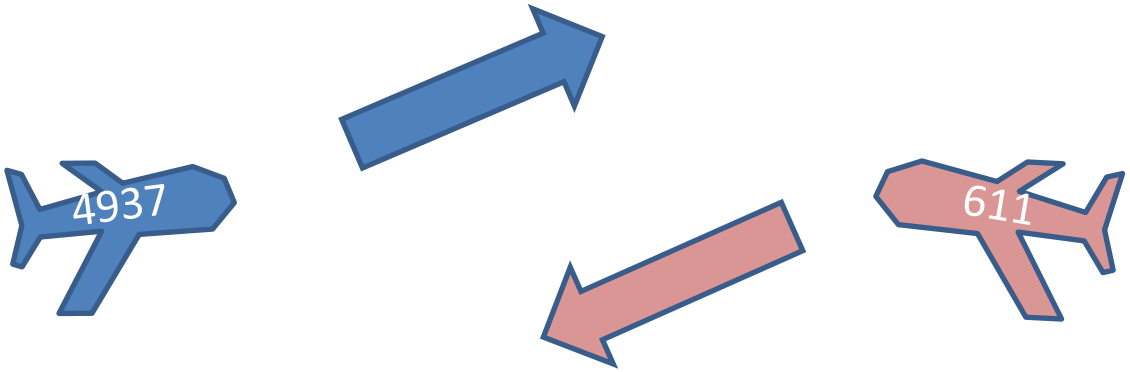
事故の経緯

2002年　ドイツユーバーリンゲン



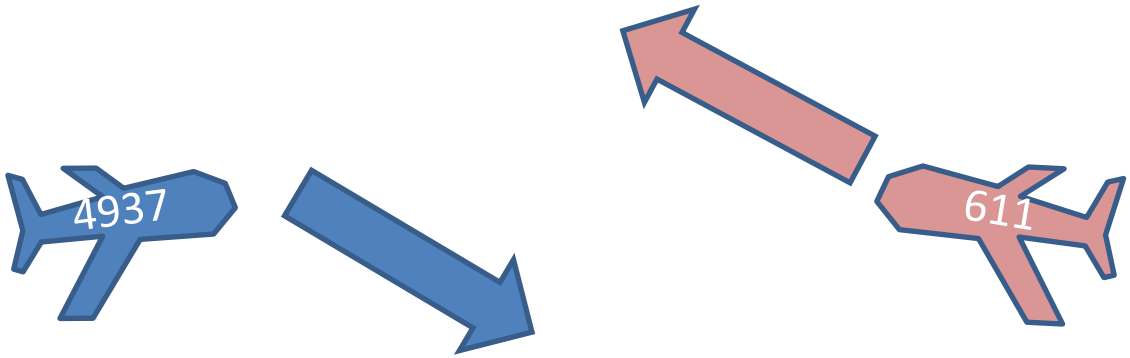
事故の経緯

機体搭載TCAS（衝突防止装置）の指示



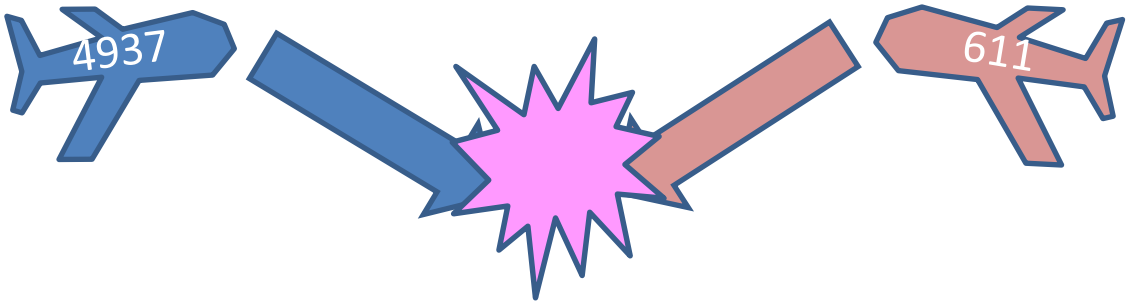
事故の経緯

地上管制官(ATC)の指示



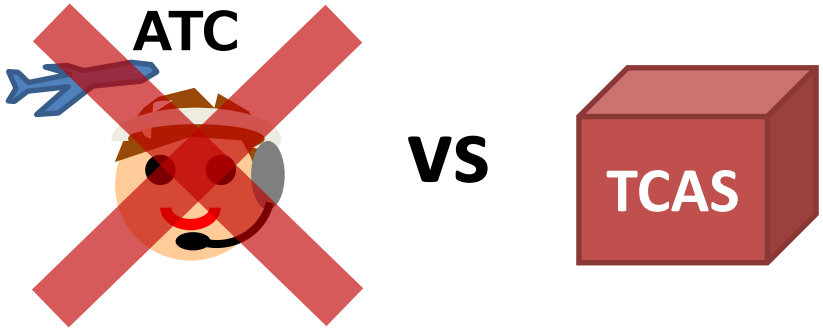
事故の経緯

パイロットの動作



事故後の対策

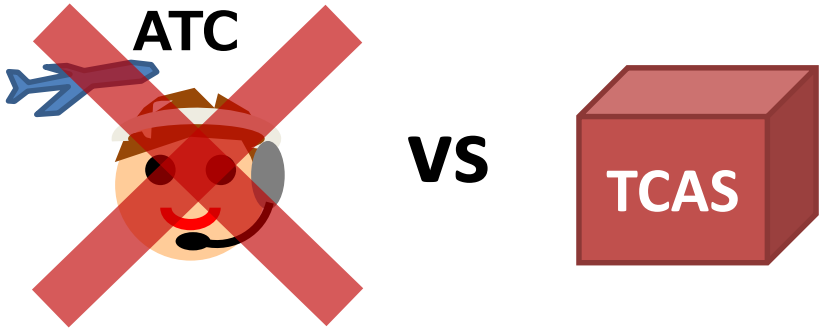
もし、管制官の指示とTCASの指示が異なる場合は、TCASの指示に従うこと。



事故の原因は管制官のヒューマンエラーと分析された。
ヒューマンエラーを防止するために、
コンピュータの指示を優先せよという対策。

事故後の対策

もし、管制官の指示とTCASの指示が異なる場合は、TCASの指示に従うこと。

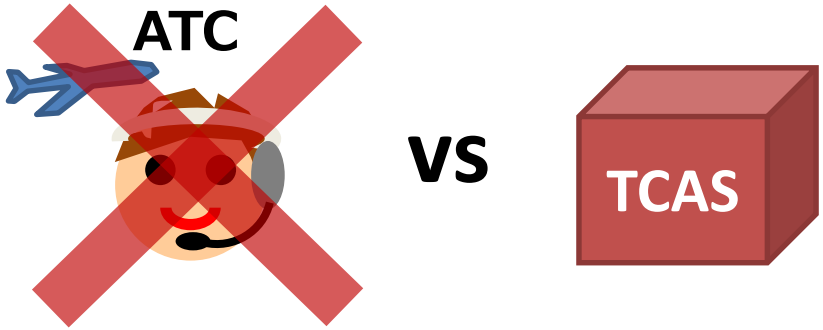


事故が起こると、「ヒューマンエラー」が原因と断定されるケースは非常に多い。

しかし、本当にそうなのか？← STAMPの発想

事故後の対策
いくつかの問題点

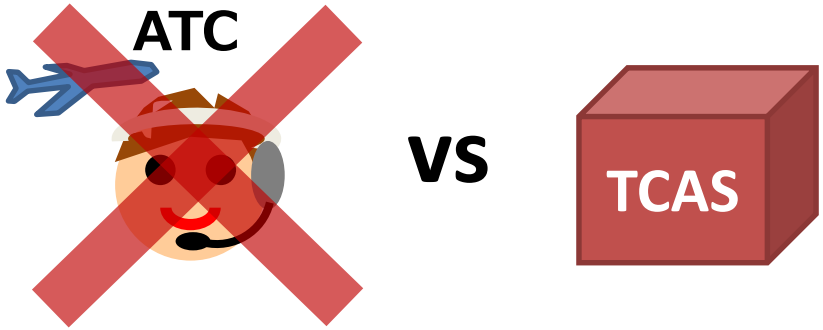
もし、管制官の指示とTCASの指示が異なる場合は、TCASの指示に従うこと。



管制官が先に衝突を予見し指示。
後からTCASが逆の指示。
⇒　両パイロットは途中から飛び方を変える？

事故後の対策
いくつかの問題点

もし、管制官の指示とTCASの指示が異なる場合は、TCASの指示に従うこと。

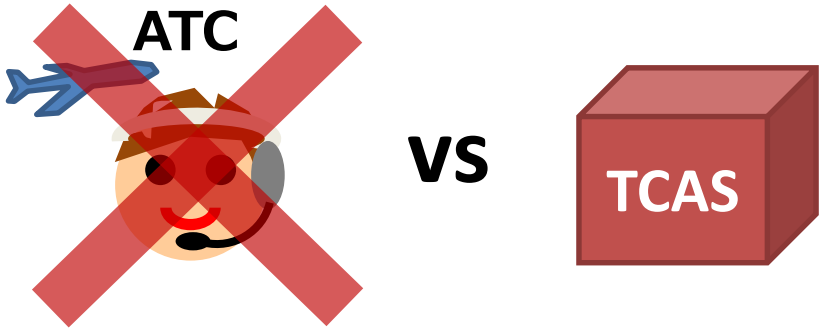


片方のTCASが故障していたとき

⇒　もう片方のパイロットはどっちを信じる？

事故後の対策
いくつかの問題点

もし、管制官の指示とTCASの指示が異なる場合は、TCASの指示に従うこと。



管制官の指示がシステム全体として適切

⇒　パイロットはそれでも管制官を無視する？

ありがちな発生要因のSTAMP分析

ハザード：空中衝突

Control Action	提供されないとハザード	提供されるとハザード	開始タイミングが悪いとハザード	終了タイミングが悪いとハザード
管制官の指示	指示が出なければハザード	間違った指示をすればハザード	遅すぎる指示をすればハザード	早すぎる指示終了はハザード

ヒューマンエラーを原因とすると、人間はまるで元凶。
人間をシステムから排除せよという結論に陥ってしまう。

ありがちな発生要因のSTAMP分析

ハザード：空中衝突

Control Action	提供されないとハザード	提供されるとハザード	開始タイミングが悪いとハザード	終了タイミングが悪いとハザード
TCASの指示	指示が出なければハザード	間違った指示をすればハザード	遅すぎる指示をすればハザード	早すぎる指示終了はハザード

TCASの設計ミスを原因とすると、TCASはまるで元凶。
TCASをシステムから排除せよという結論に陥ってしまう。

ありがちな発生要因のSTAMP分析

ハザード：空中衝突

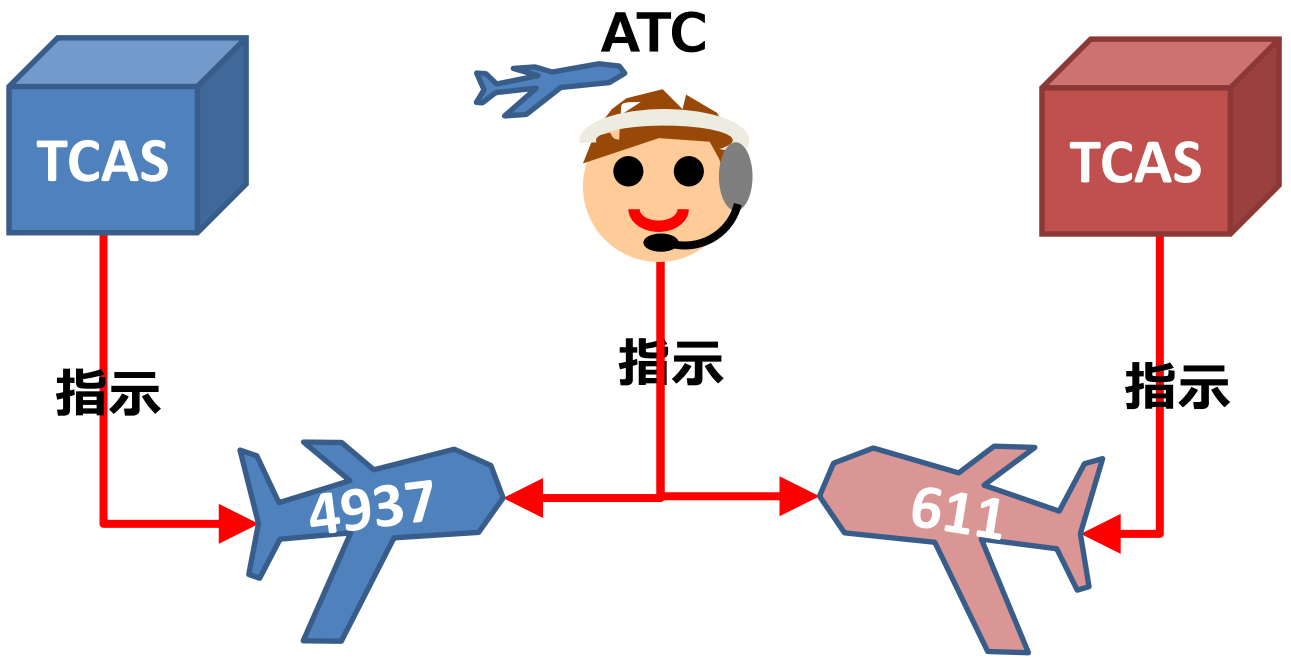
Control Action	提供されないとハザード	提供されるとハザード	開始タイミングが悪いとハザード	終了タイミングが悪いとハザード
管制官の指示	指示が出なければハザード	間違った指示をすればハザード	遅すぎる指示をすればハザード	早すぎる指示終了はハザード
TCASの指示	指示が出なければハザード	間違った指示をすればハザード	遅すぎる指示をすればハザード	早すぎる指示終了はハザード

どちらの分析にも、あまり意味は無い
事故を誰かのミスのせいにしても、得るものは少ない

システム的な分析

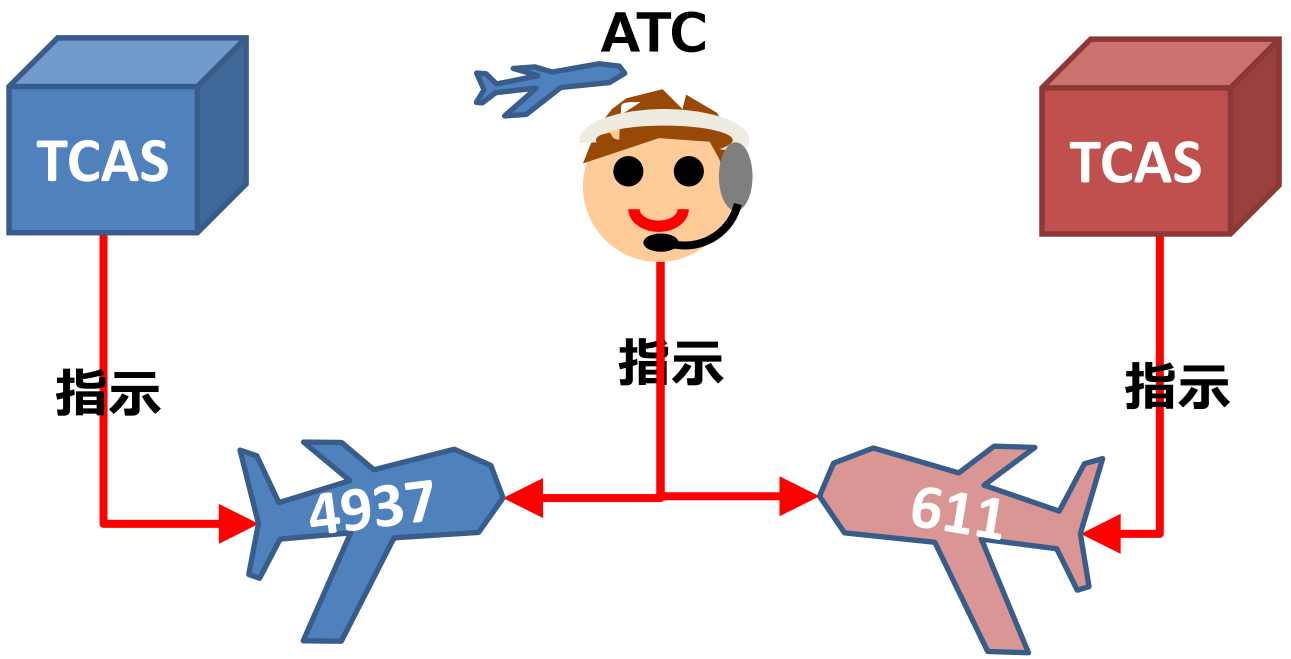
制御構造の分析によって、
事故の発生メカニズムを
システム的にとらえる

制御構造



２つのControlled Processに対して
３つのControllerが競合する

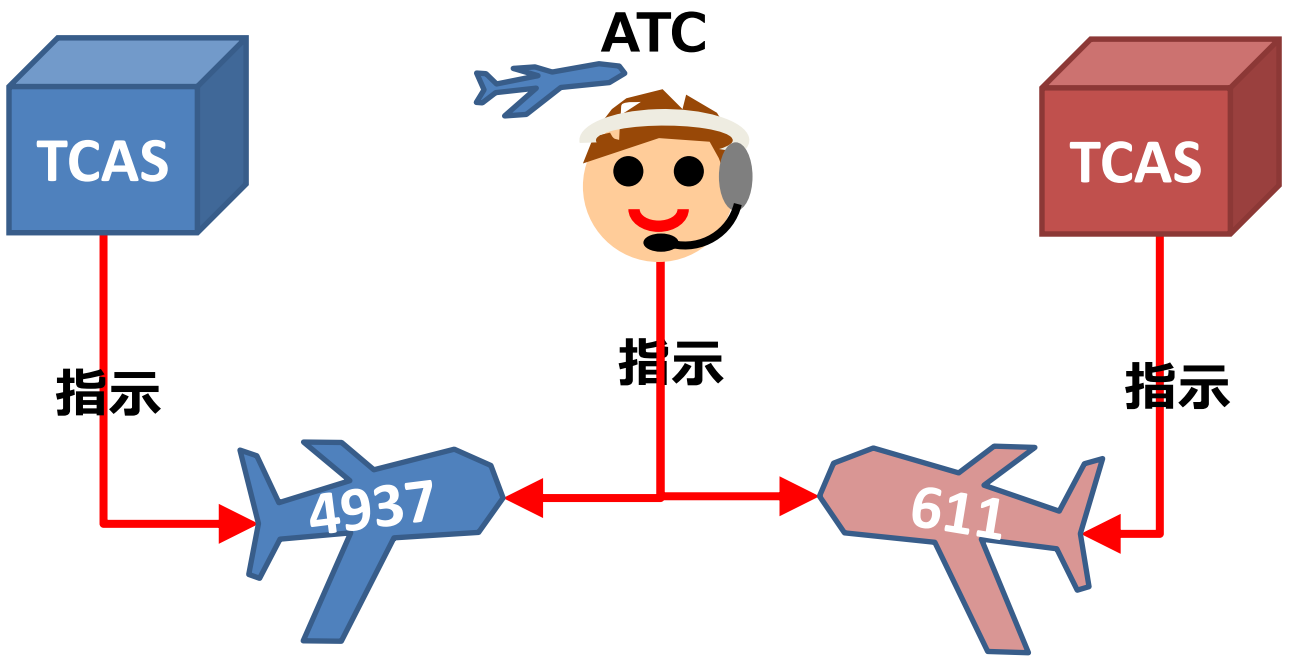
制御構造



STAMPの発想：

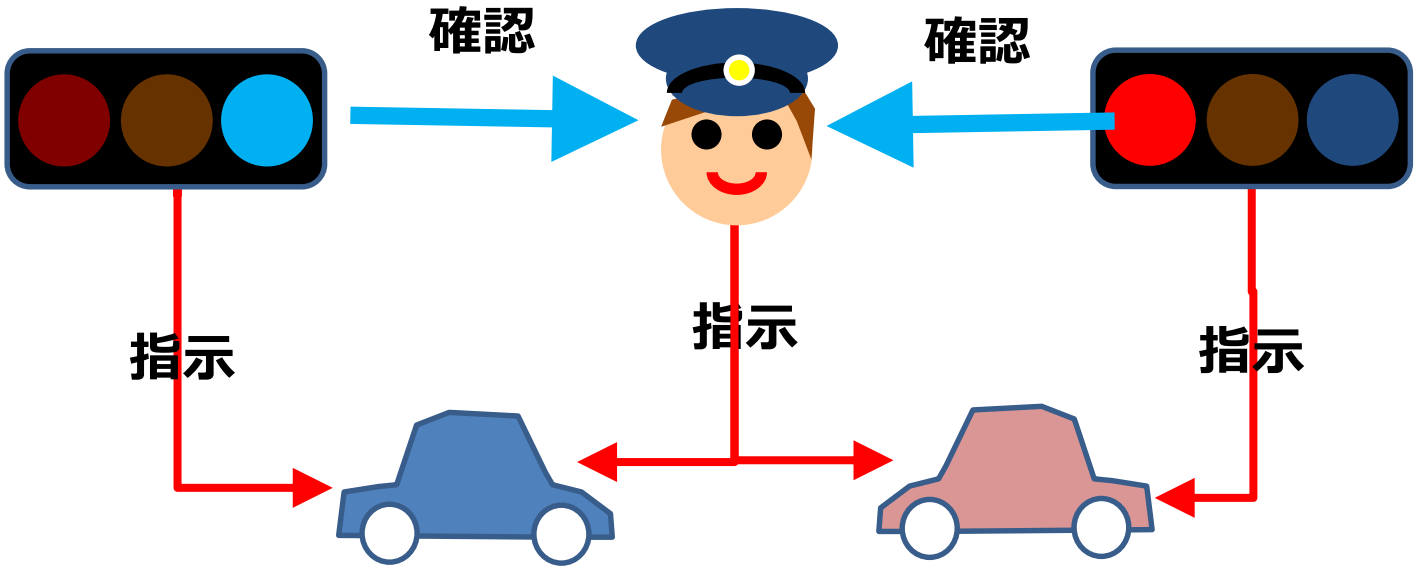
事故は、この制御構造から創発的に起こる

制御構造



制御構造から**創発的**に発生する事故とは？

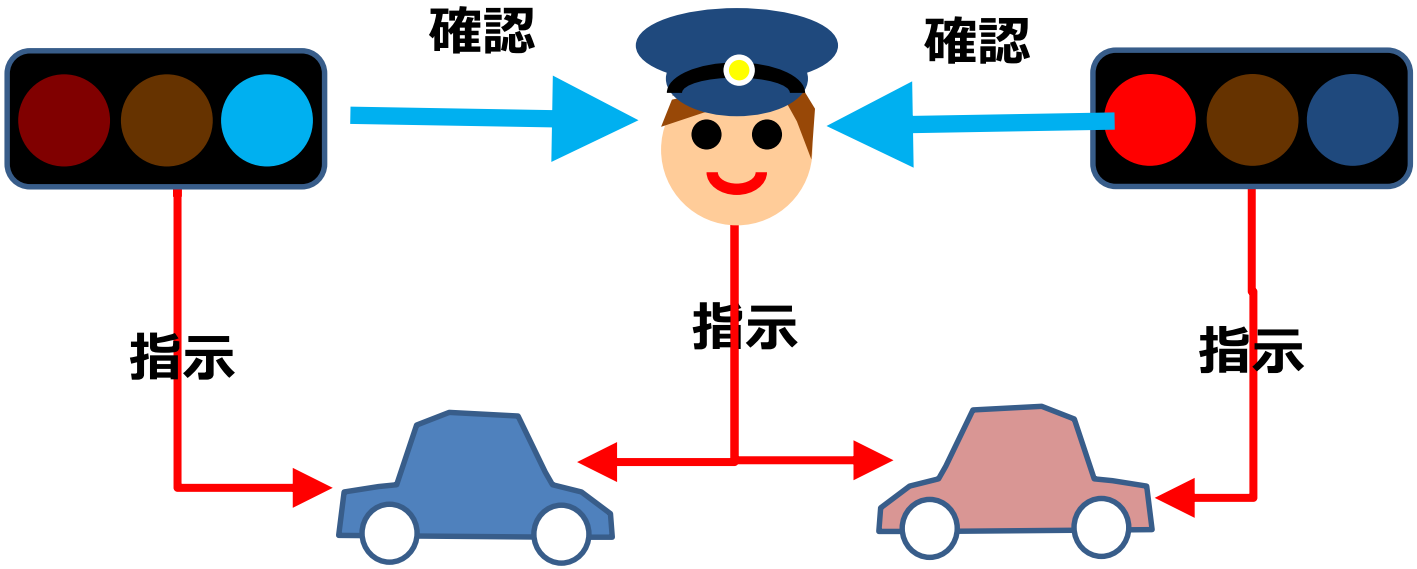
似た構造を持つシステム



成功要因：「確認」

交差点の交通整理

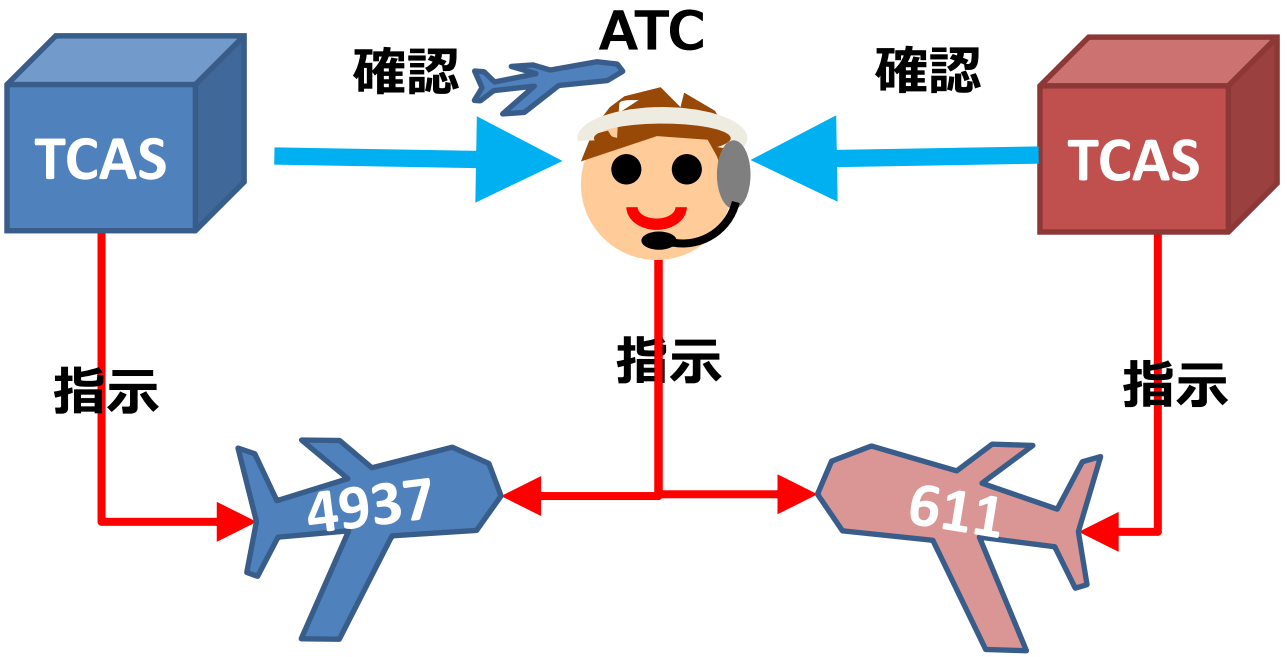
似た構造を持つシステム



確認があるだけで、人間中心になり、効果が高い

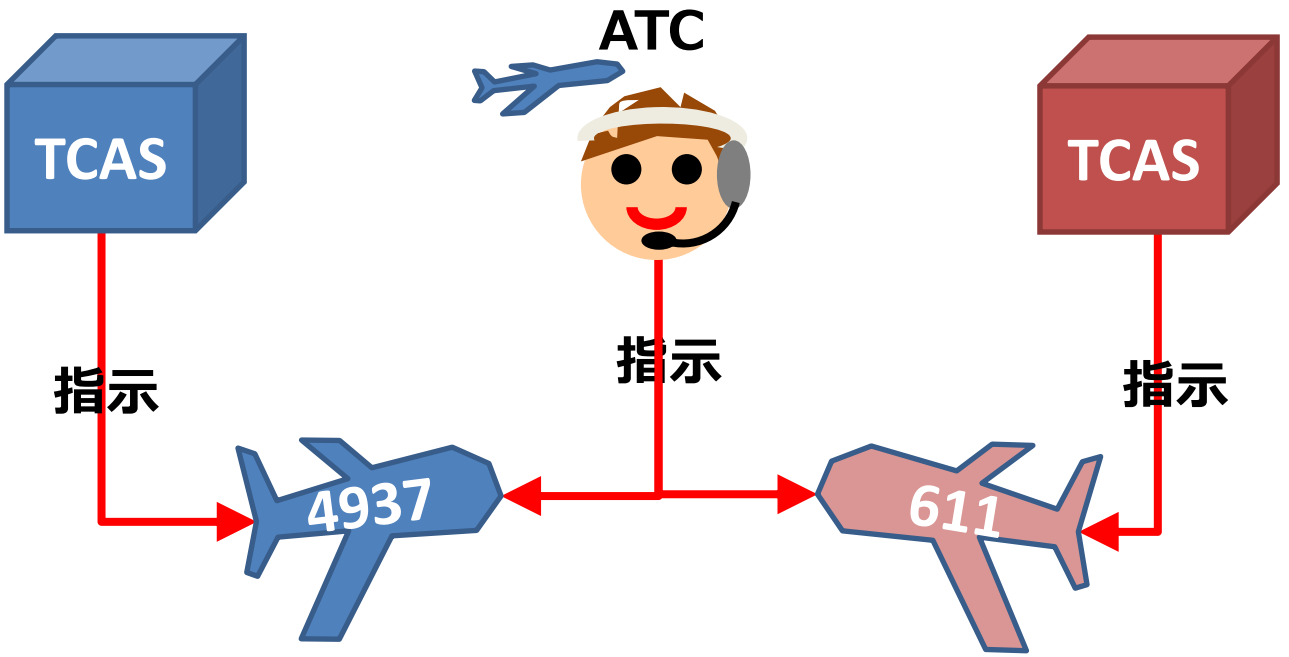
交差点の交通整理

制御構造



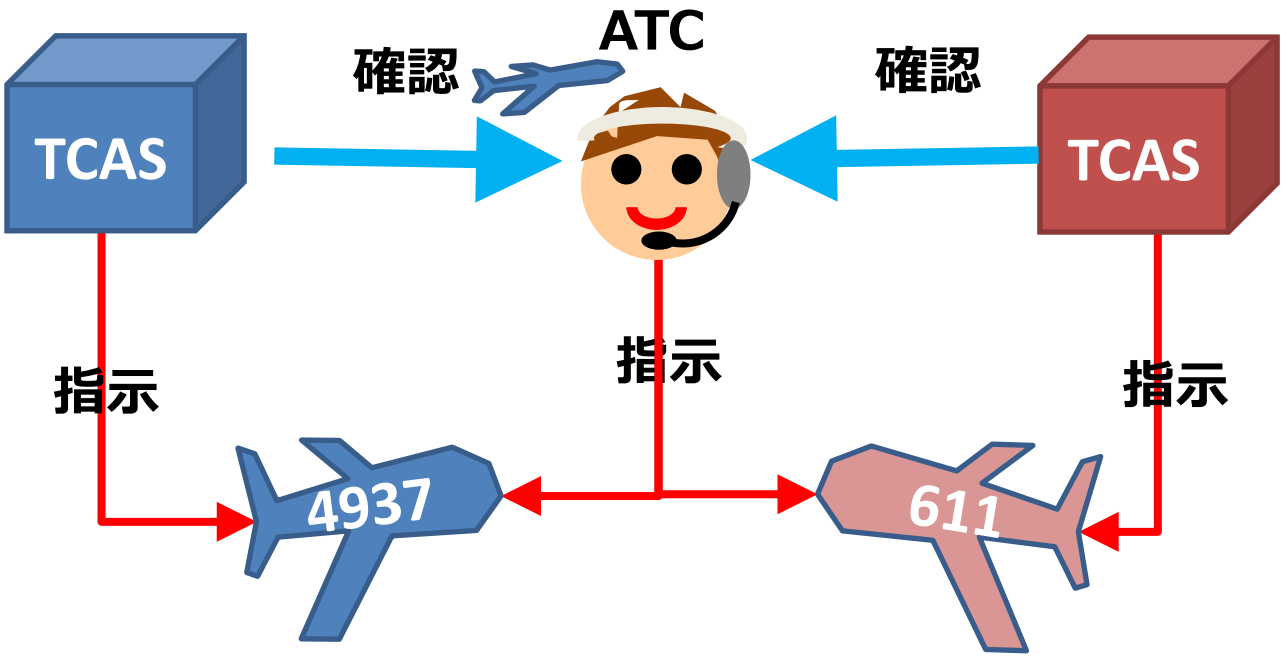
確認が無いため、機械中心となり、管制官は
ヒューマンエラー源となる。

制御構造



確認が無いことが、ハザード要因
つまり、これが、構造から創発する事故原因

制御構造



制御構造を変えれば、構造的な安全性向上

構造的な安全性 ≒ 本質安全

まとめ

- STAMPは、事故モデルを「制御構造図」で表現。
- 制御構造の解析能力が、STAMPに必要な素養
  - 制御構造の解析 = System Engineering
  - 制御構造の解析 = Emergent Propertyの導出
- 制御構造の解析により、「ヒューマンエラー」排除の思想を超え、人間中心の安全性を実現可能
- 制御構造の解析が、構造的安全性≒本質安全への道
- 制御構造の解析により、システムの成功要因を識別。エラーの組み合わせをしらみつぶしに排除せずとも、安全性を一挙に向上できる ⇒ $O(n^2)$からの脱却