2025 年 4 月 15 日更新
独立行政法人情報処理推進機構
デジタル基盤センター

STAMP および STAMP ツールの Tips

内容

■便利な操作方法

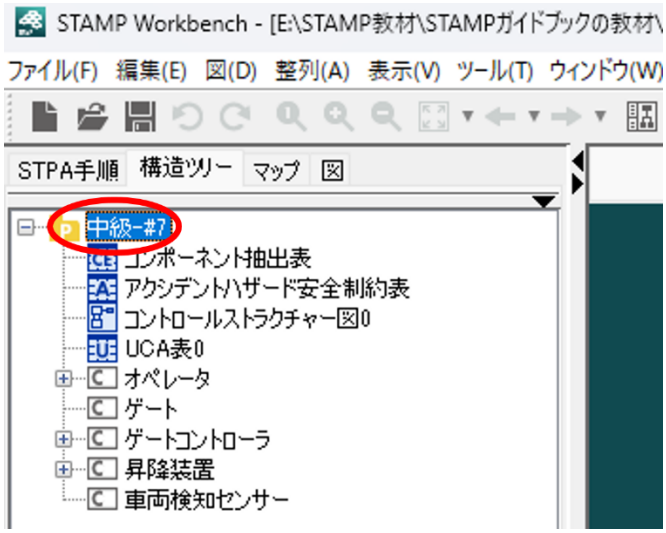
図や表の文字サイズを変更する

ツールとして常に有効にする方法ではなく、プロジェクトファイル毎の設定です。

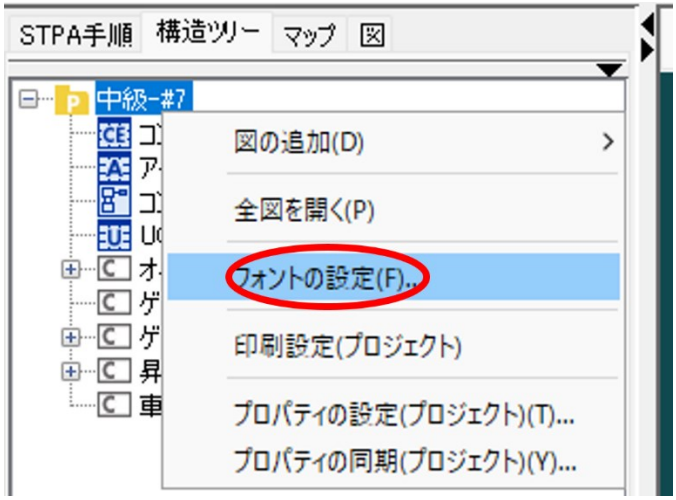
1. STAMP Workbench で、文字サイズを変更したいプロジェクトファイル（.stmp ファイル）を開く。

2. 左上プロジェクトビューのタブに STPA 手順/構造ツリー/マップ/図 があります。「構造ツリー」タブを選択。

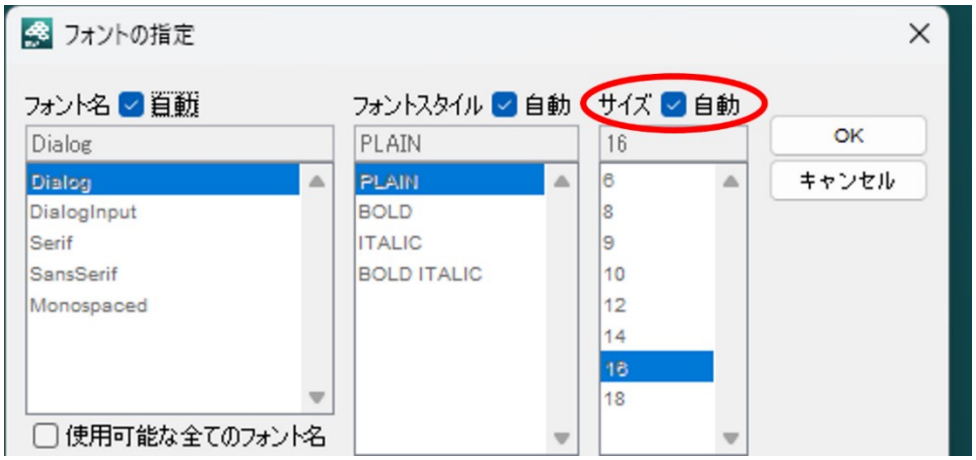
3. プロジェクトファイル名を選択し、コンテキストメニューを開く（右クリック）。
下図では「中級-#7」がプロジェクトファイル名。



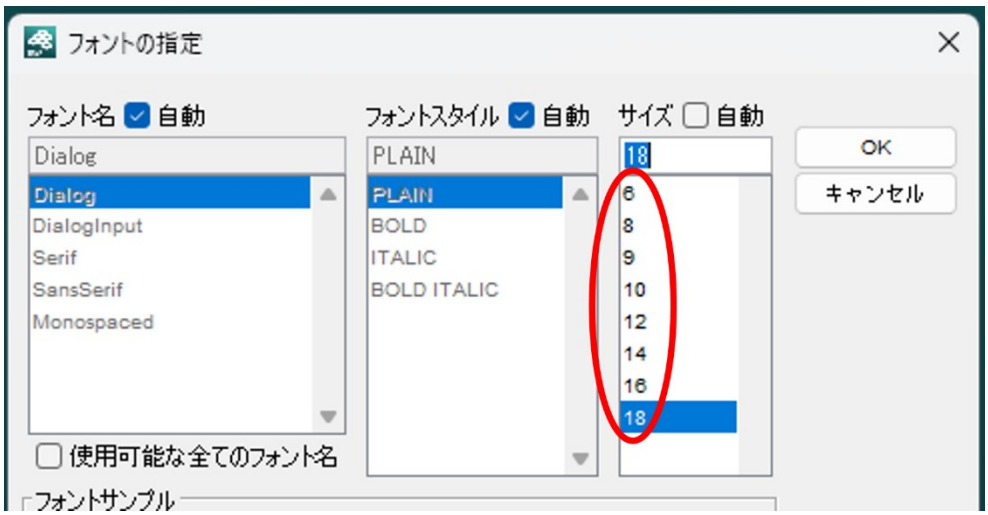
4. プルダウンメニューから、「フォントの設定」を選択。



5. 「フォントの指定」ダイアログで、「自動」が選択されているので、オフにする。



6. 文字フォントのサイズを数字で選択。デフォルトは 16。

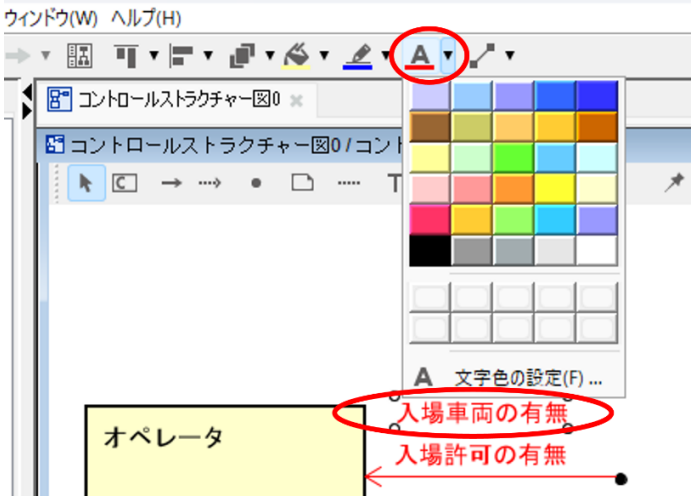


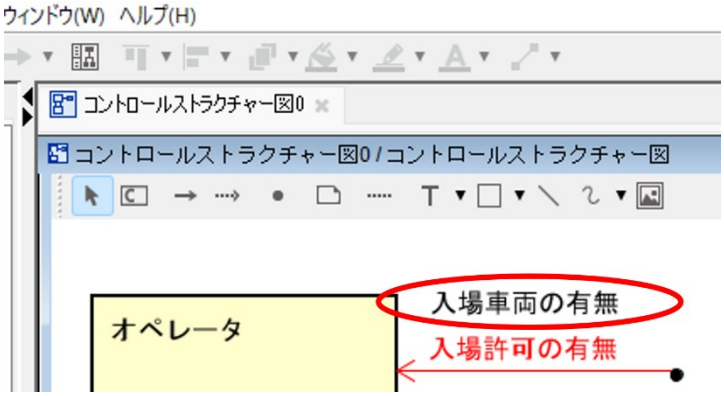
設定できるサイズは 6 から 18 まで。
OK を押して確定する。
選択したフォント指定はすべての図や表に適用される。別の方法を使って、CS 図上の一部の文字列のみ文字色やフォントを変更することもできる。

目立つように一部の文字だけ文字色を変更する

コントロールストラクチャー図（CS 図）において一部の文字列のみ文字色やフォントを変更することができます。全ての表で一部の文字単位で文字色を変更する機能はありません。

CS 図やコントロールループ図において、CA、FB、input、output など色を変更したいコンポーネントや文字列を選択して、ツール画面にある帯状のツールバーで「文字色の設定」を押して色を選択する。変更できるのは文字色とフォント種類で、文字サイズは変更できない。





変更できない例：文字単位で色を変更

あいうえお

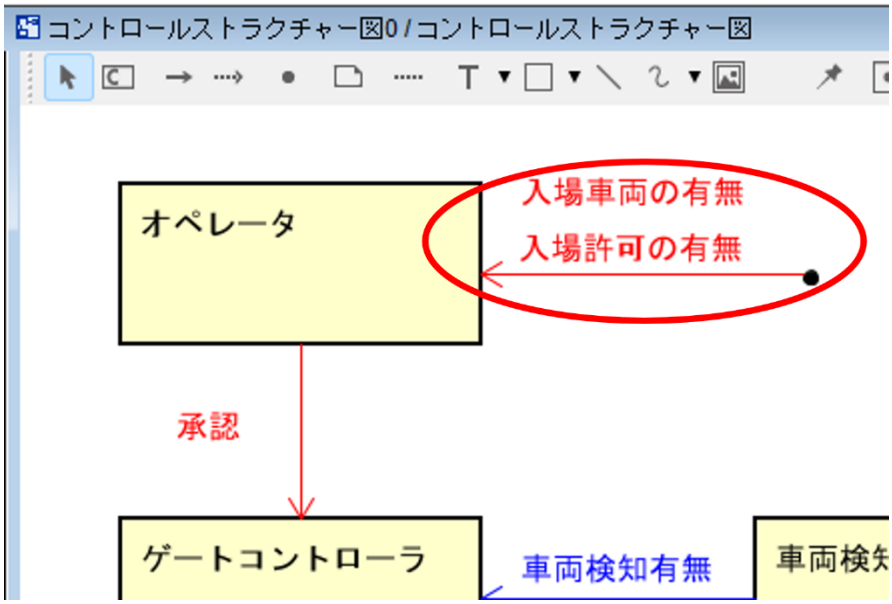
変更できる例：文字列全体の色とフォントの変更

あいうえお

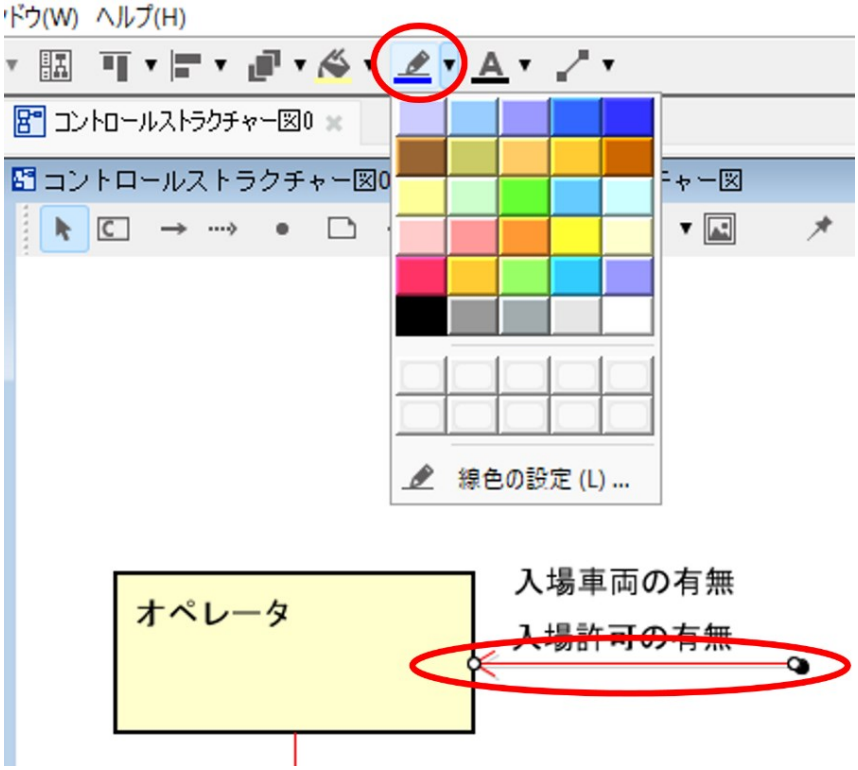
input/output 線の操作方法

STAMP のコントロールストラクチャー図（CS 図、CSD）において、input/output は、外界からシステム内のコンポーネントへの入力/システム内のコンポーネントから外界への出力を意味します。
CS 図、あるいはコントロールループ図（CL 図、CLD）において、コンポーネント間の線は制御（CA）またはフィードバック（FB）であり、input/output は入力元や出力先のコンポーネントが現れません。
但し、「外界モデル」の項に記したように、外界の何との input/output かを示したいときには外界モデルというコンポーネントを配置して、その外界モデルコンポーネントとシステム内コンポーネントの間を input/output 線で結ぶ方法もあります。
その場合、STAMP Workbench では input/output 線が CA 線（赤い矢印線）または FB 線（青い矢印線）で結ばれるので、線の色を黒に変更すると良いです。
CA、FB、input/output の違いを線の色で明確に区別することをお勧めします。
この対策はツールが自動的にサポートすることが望ましいですが現状サポートできていません。

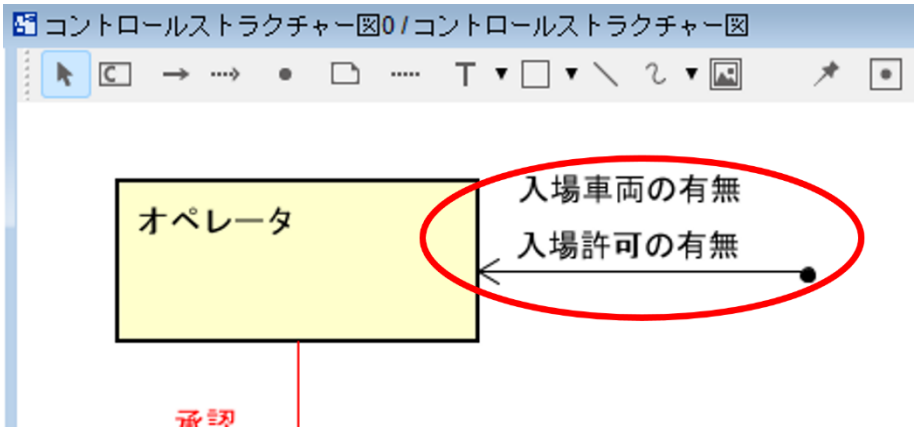
下図では、input の文字も線も CA と同じ赤になっているので、CA と勘違いし易い。



変更したい線を選択すると線色変更のボタンが有効になる



下図では外界からの input の文字色と線色の両方を黒に変更した。CS 図上で CA と input の違いが明確になった。

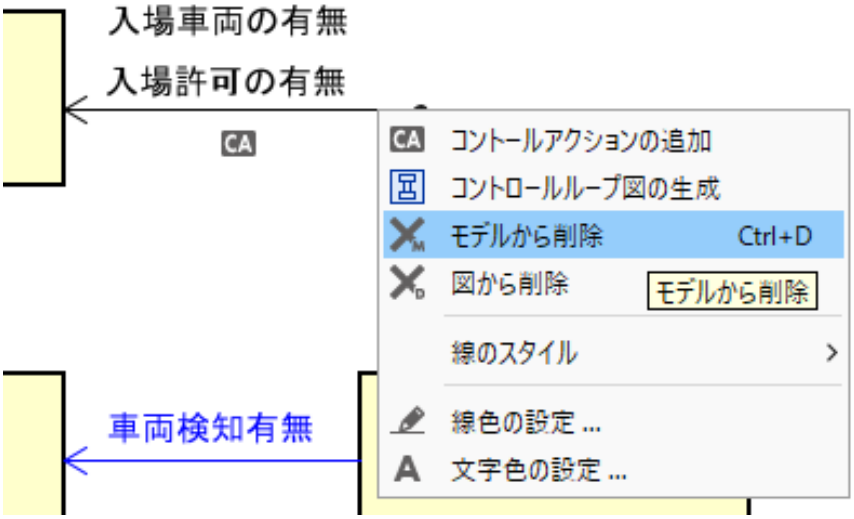


図から削除／モデルから削除

コントロールストラクチャー図（CS 図）を編集する際、コンポーネントや CA/FB 線などを削除する場合、削除対象を選択してからコンテキストメニューを開く（右クリックする）と、「モデルから削除（Ctrl＋D)」と「図から削除（Delete)」を選択できる。

「モデルから削除（Ctrl+D)」を選択することをお勧めします。

「図から削除（Delete)」とすると見かけ上は見えなくなりますが、モデルとして残っている（コンポーネント抽出表にも残っている）ので、後で何らかの編集をしているときに図に復活することがあり戸惑うことが有り得ます。不要と判断したら「モデルから削除（Ctrl+D)」を選択することをお勧めします。



Excel に変換して列を増やす

UCA 表や対策表に列を増やしたいことがあります。例えば、

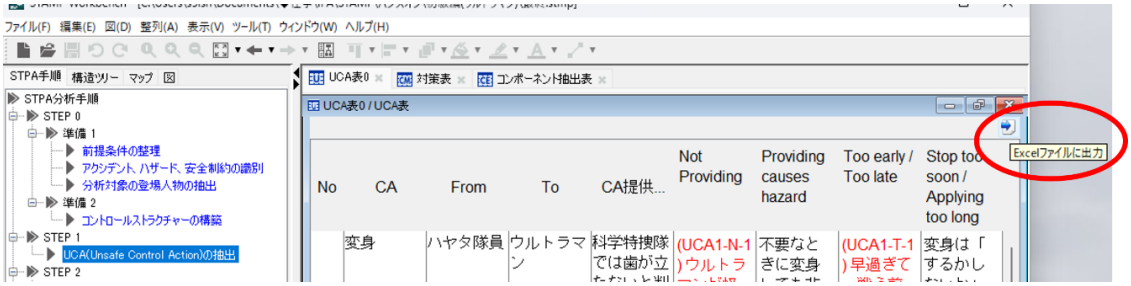
- UCA 表にメモを残したい。UCA を考えるときには原因を考えないようにするが、思いつくことはある。あとのステップのために思いついたことをメモしておきたい。
- 対策表に、自社（自プロジェクト）独自の ID などの情報をつけておきたい。

等々、STAMP Workbench への機能拡張要望が多々あります。STAMP Workbench は汎用的かつ基本的な機能を実装しており、とりあえずたいていの場合は事足りることを想定しているので、IPA による機能拡張予定はありません。このような要望については、次の 2 通りの対応策があります。

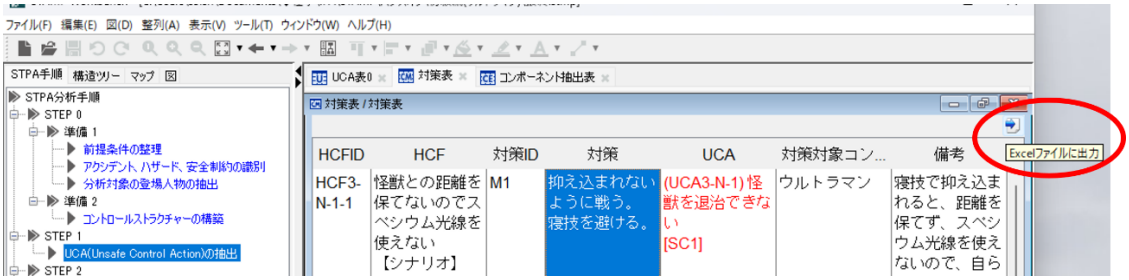
【対応策 1】
利用者様自身が（IPA がオープンソースとして公開している）ソースコードを改編して機能追加することも可能です。モデリングツール一般に精通していない技術者様でも、実は、列を追加する程度の改編は然程ハードルの高いものではありません。プログラミングの腕に覚えのある方ならば、ソースコードにザックリ目を通していただければお判りいただけるかと思います。ソースコード改編後のビルド方法も公開しています。改編して利用されている事例は国内外から報告されています。

【対応策 2】
STAMP Workbench で作成した表を Excel 出力して、Excel 上で列追加する。

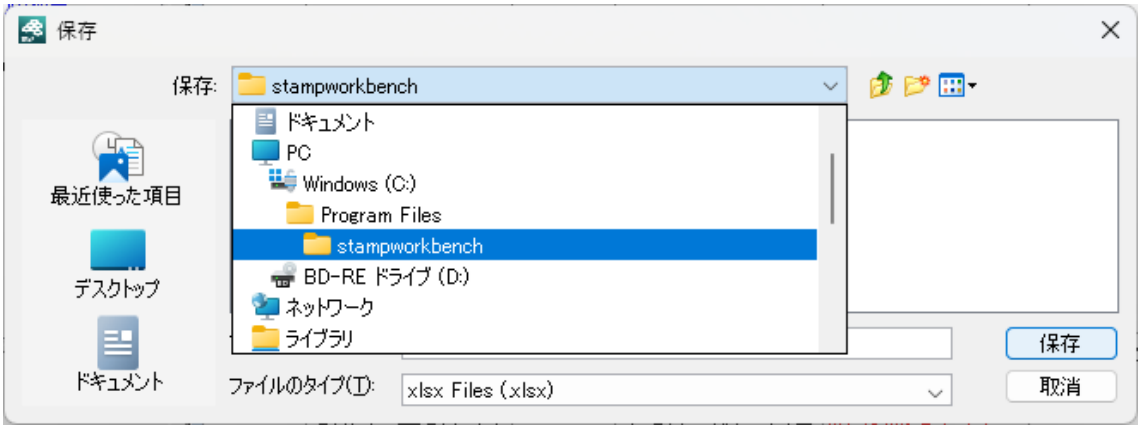
(1) UCA 表や対策表ウィンドウの右上にある「右向き矢印」を押す。
下図は UCA 表の場合。



対策表の場合も同じ。



(2) 「保存」のウィンドウを開いたら、保存先を選択して「保存」ボタンを押す。



保存先に Excel 表（.xlsx ファイル）が保存されるので、Excel を起動して任意に列追加などを行う。

注意事項

STAMP Workbench から Excel へのエクスポート機能は有るが、Excel から STAMP Workbench へのインポート機能は無い。
Excel で編集した内容を自動的に STAMP Workbench に取り込むことはできない点に注意のこと。

デフォルトの保存先フォルダーは Program Files フォルダーなので書き込みできません。必ず、プロジェクトファイル（.stmp ファイル）と同じフォルダーなど、書き込み可能なフォルダーを選択してください。

CS 図の CA,FB に番号を付ける

CS 図を見て議論するときに CA,FB に番号があると話が早いです。
CA,FB の番号振り直しは次のようにすると簡単です。

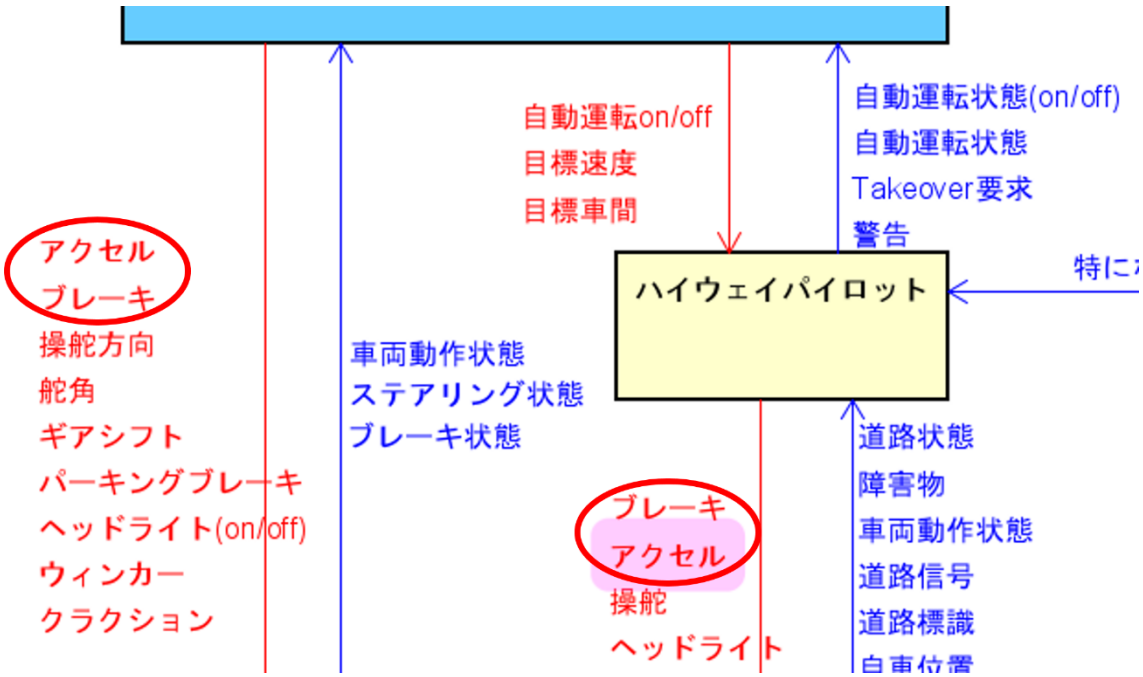
1. CS 図で直接 CA,FB の名前を編集して"CAn"のように適当に番号を振る。
2. UCA 表で CA を"CAn"の n に合わせてドラッグ＆ドロップで上下に移動する。
   CA を押して掴み、挿入したい横線のところまで上下にマウスカーソルを移動して放す。
   または、
3. UCA 表の最左列の CA 番号に合わせて、名前に付けた"CAn"の n を修正する。

ツールへの要望

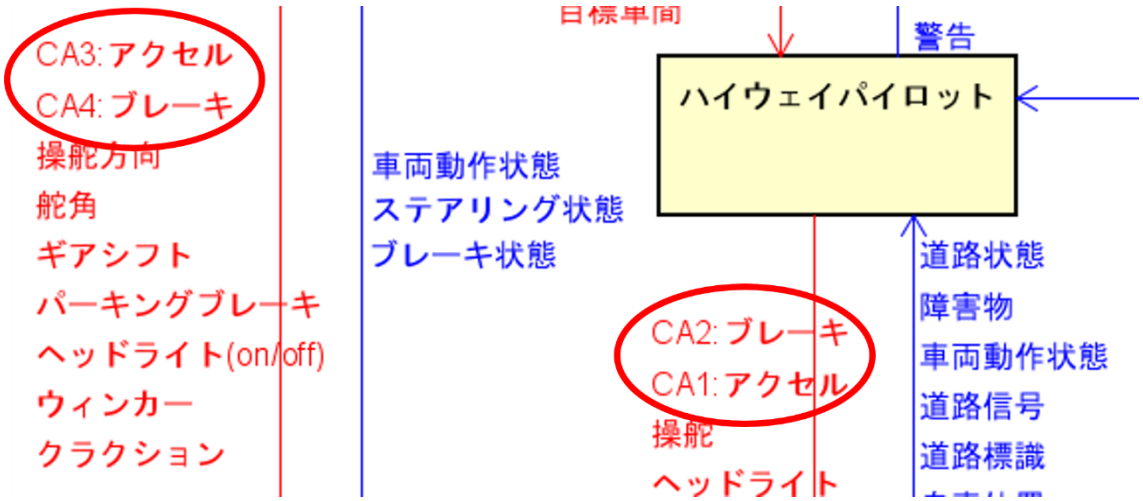
UCA の番号は UCA 表で整理できるが、FB の番号は CS 図かコンポーネント抽出表でやるしかないので不整合が出そうな気がする。できることなら、（一般的には CA と FB は対になるので）コントロールループを成す CA と対応付く FB が同じ番号になると良いが、自動的にナンバリングするのは難しそう。せめて、番号の重複を無くすくらいはできると良い。

この対策はツールが自動的にサポートすることが望ましいが現状サポートできていません。

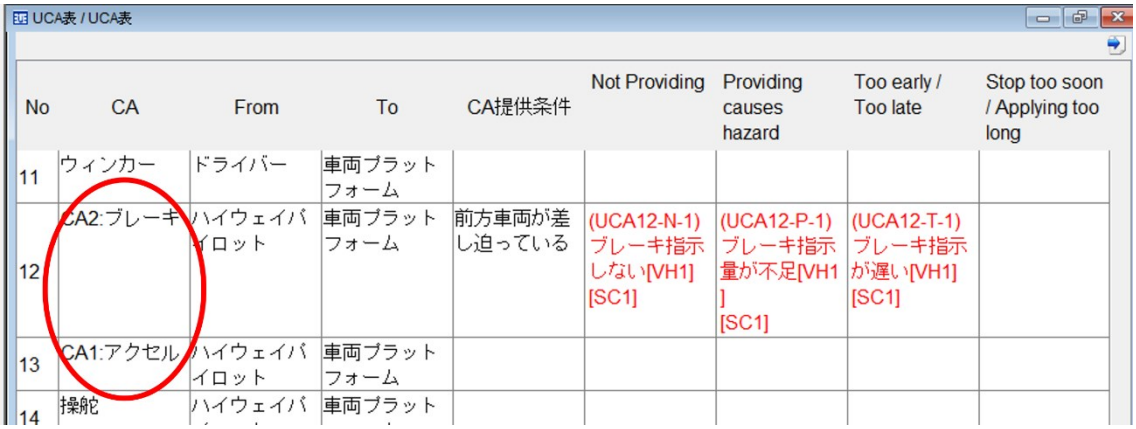
下図のように「アクセル」や「ブレーキ」といった同名の CA があるときなど、レビュー時に言葉だけでは区別が面倒なので CS 図に番号が表示されると便利。



CS 図の中で CA1、CA2、・・のように番号を振る。



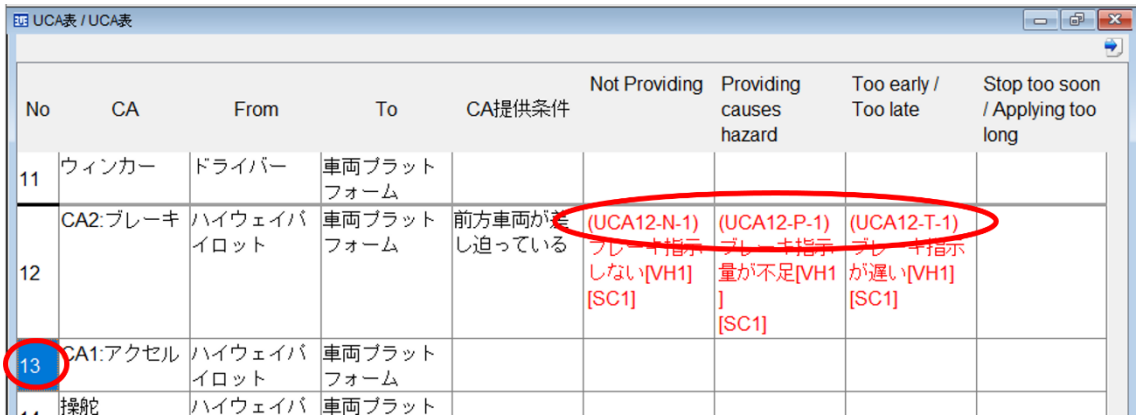
UCA 表で順番が上下逆なのが気になる。



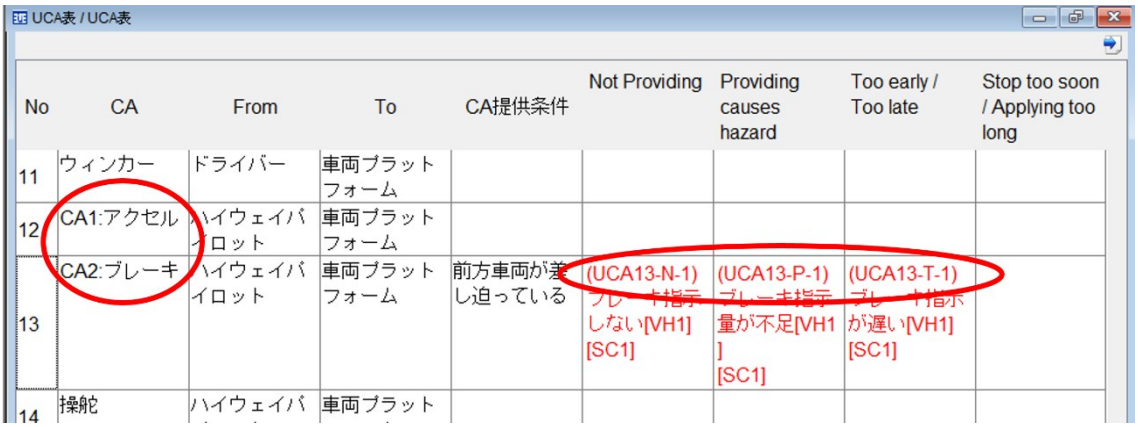
UCA表 / UCA表

No	CA	From	To	CA提供条件	Not Providing	Providing causes hazard	Too early / Too late	Stop too soon / Applying too long
11	ウィンカー	ドライバー	車両プラットフォーム					
12	CA2:ブレーキ	ハイウェイパイロット	車両プラットフォーム	前方車両が差し迫っている	(UCA12-N-1) ブレーキ指示しない[VH1] [SC1]	(UCA12-P-1) ブレーキ指示量が不足[VH1] [SC1]	(UCA12-T-1) ブレーキ指示が遅い[VH1] [SC1]	
13	CA1:アクセル	ハイウェイパイロット	車両プラットフォーム					
14	操舵	ハイウェイパ	車両プラット					

UCA 表で No 欄の番号を選択してドラッグし、移動したいところまでマウスカーソルを移動してドロップする。



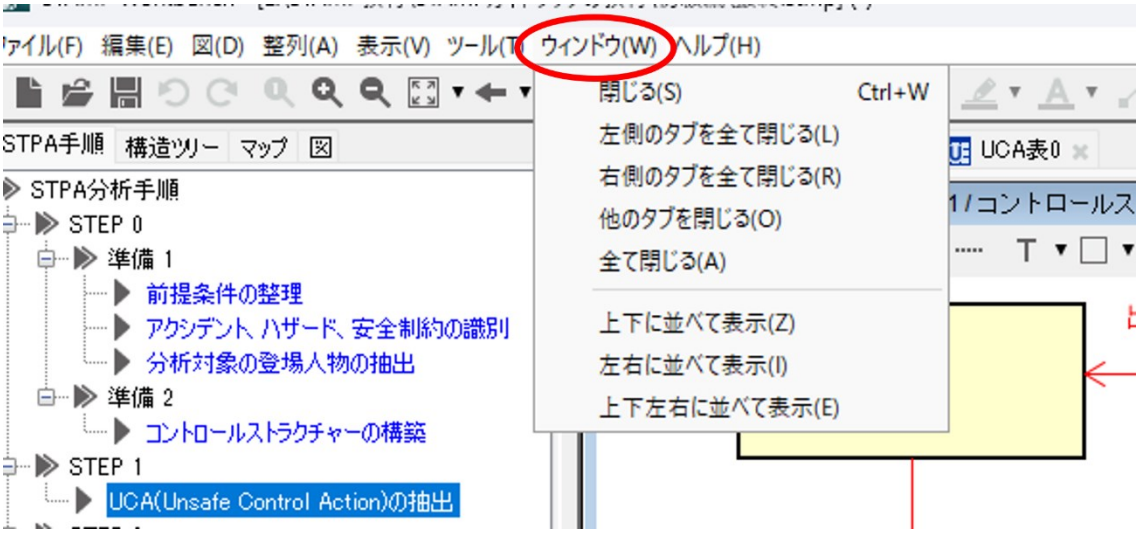
UCA 表の中の UCA 番号（UCA12-N-1 など）は自動的に変わるので、手作業での修正は不要。同時に、対策表の中の UCA 番号なども自動的に変わるので、UCA 表以外に対して手作業での修正は不要。



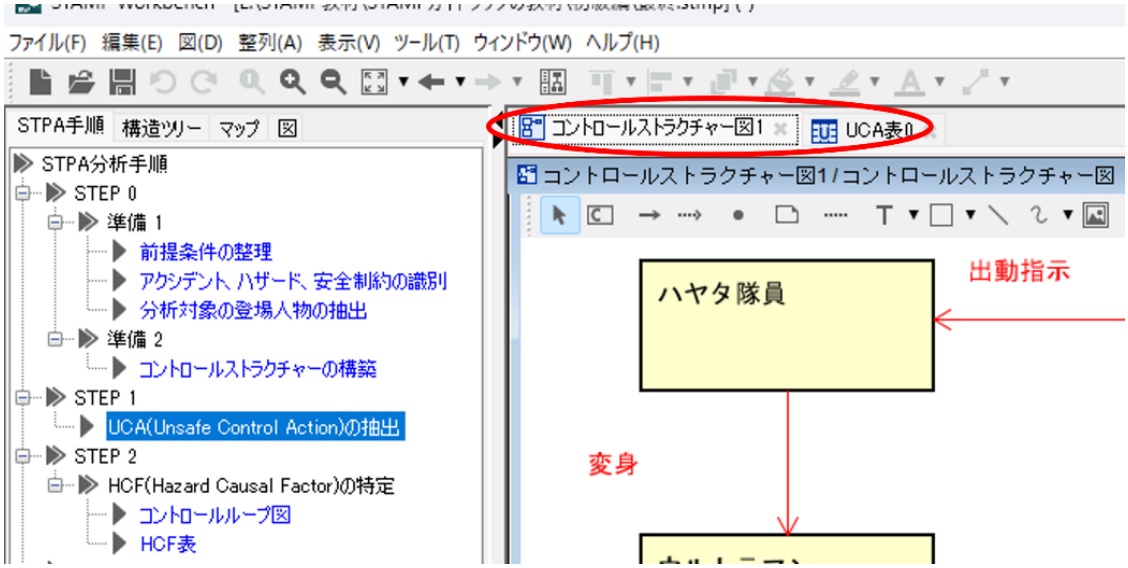
一画面に複数 window を表示する、CS 図と UCA 表を同時に見ながら UCA を考える

STAMP Workbench を用いて分析（思考）するとき、異なる画面を行ったり来たりするのは思考の妨げになるので、関係する情報を一度に見たくなります。そういうときには STAMP Workbench の「ウィンドウ」機能を用いて、1 画面内に複数の Window を表示することが有効です。

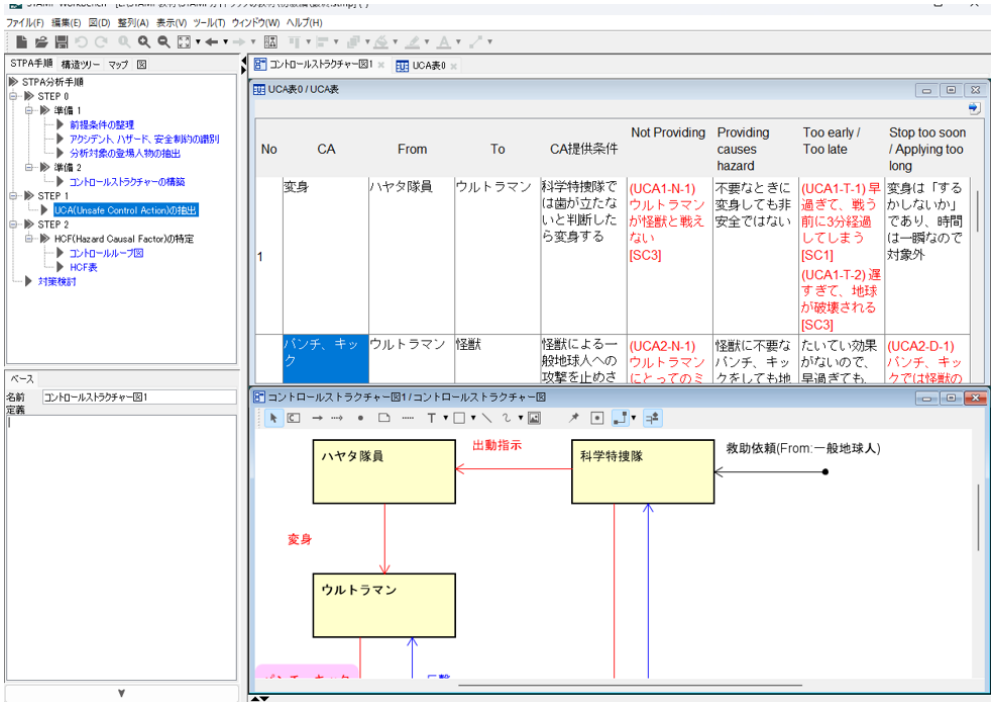
1. 同時に見たい Window を開く。



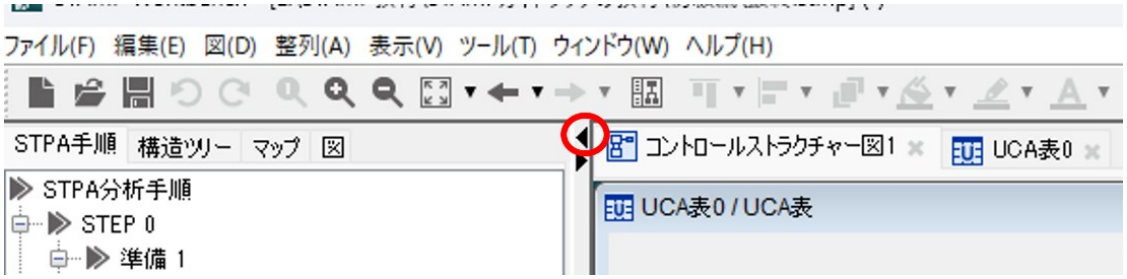
2. ツールバーの「ウィンドウ」を選択してポップアップウィンドウを開く。



3. ポップアップウィンドウで、「上下に並べて表示」、「左右に並べて表示」、「上下左右に並べて表示」から並べたい配置を選ぶ。



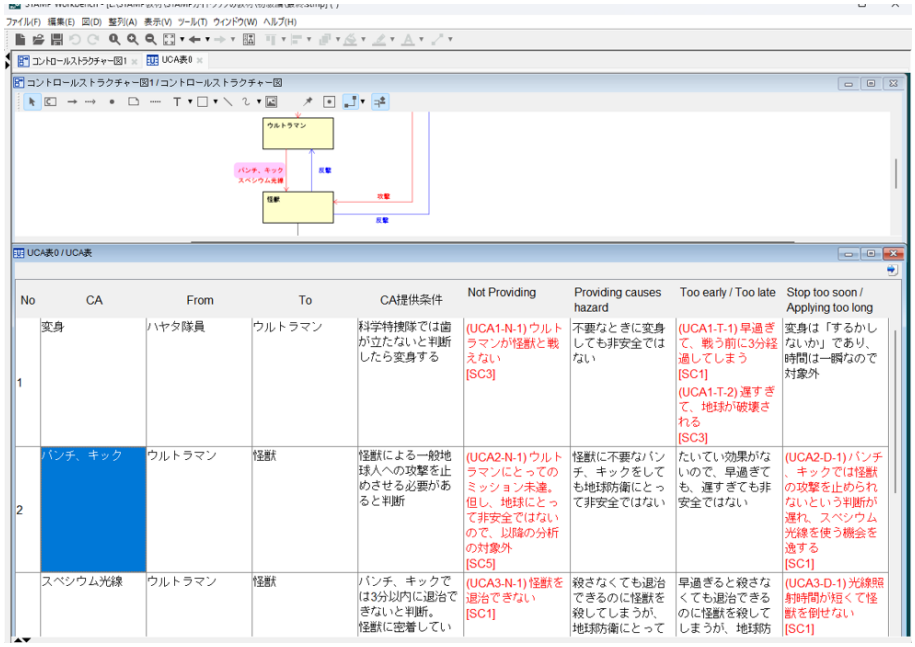
4. 画面を最大化するため、左側のプロジェクトペインを隠す。プロジェクトペインの右上の◀ボタンを押す。

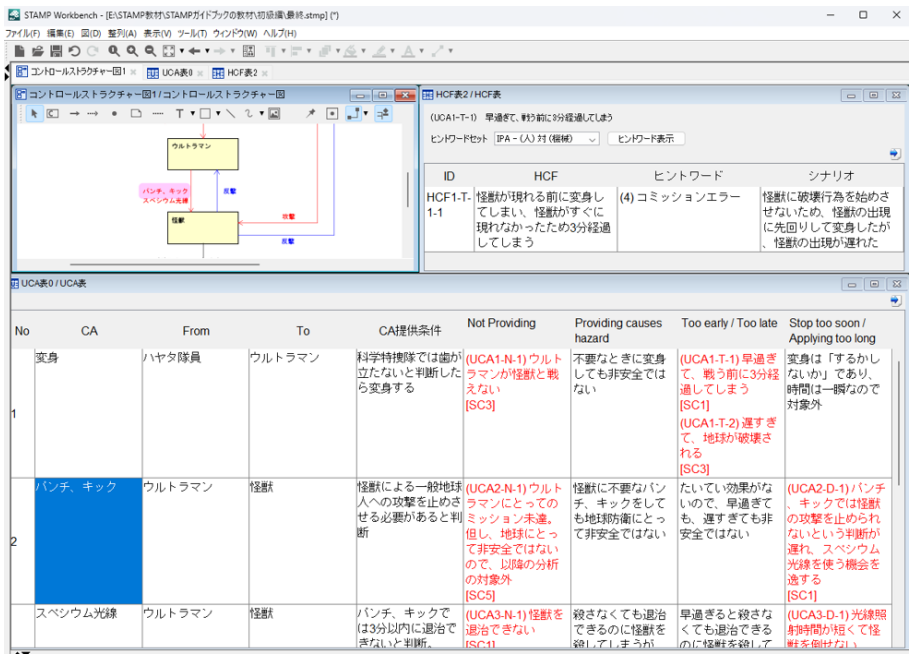


5. マルチウィンドウになったら、それぞれのウィンドウサイズを変更したり、配置を変えて、見易くする。

   下図の 1 枚目は 2 つの Window を同時に表示。2 枚目は 3 つの Window を同時に表示。

   1 枚目の図では、CS 図と UCA 表を同時に見ながら UCA を考えるときに良く使うものと思われる。この例では、CS 図の中の着目する CA に関するコントロールループのみを表示している。2 枚目の図では、コントロールループと UCA を見ながら HCF やシナリオを考えるときに有効。

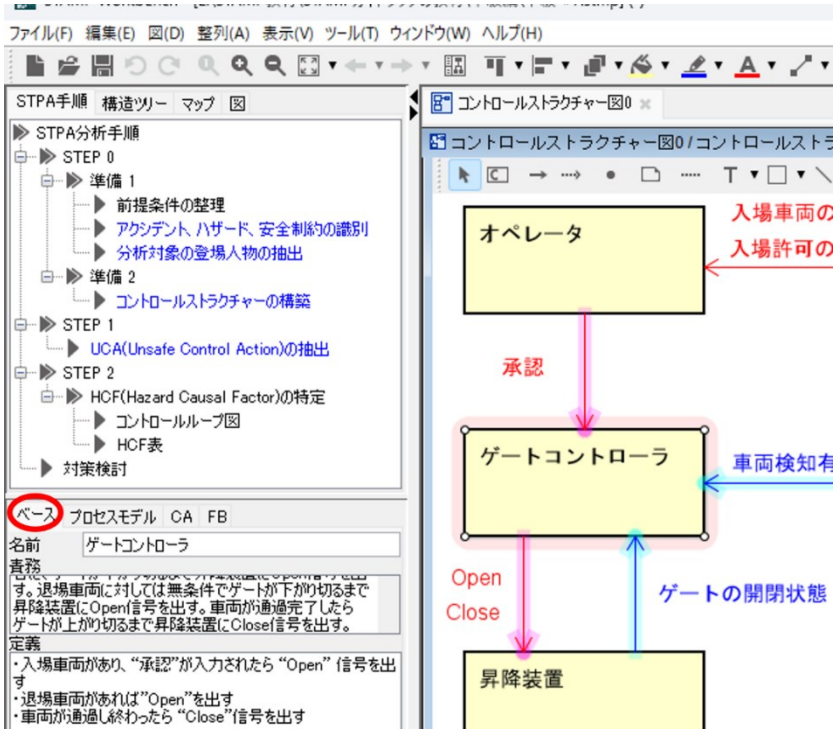




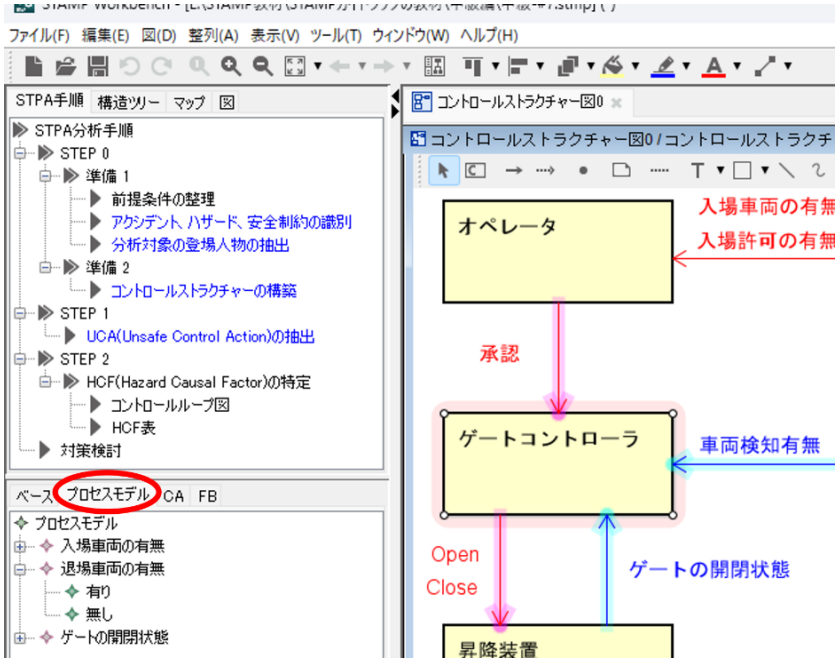
CS 図を見ながらコンポーネントの責務を確認する、プロセスモデルを入力する

CS 図を見ながら CA や UCA を考えているとき、考えている最中の対象コンポーネントの責務や input/output などを確認する場合、コンポーネント抽出表を見れば書いてあるが、画面を変えずにサッと確認したいことがある。その場合は、画面左下のペインに表示させることができる。

1. CS 図で対象コンポーネントを選択。
2. 画面左下のペインでベースタグを押す（デフォルトはベースタグの画面が表示されている）。

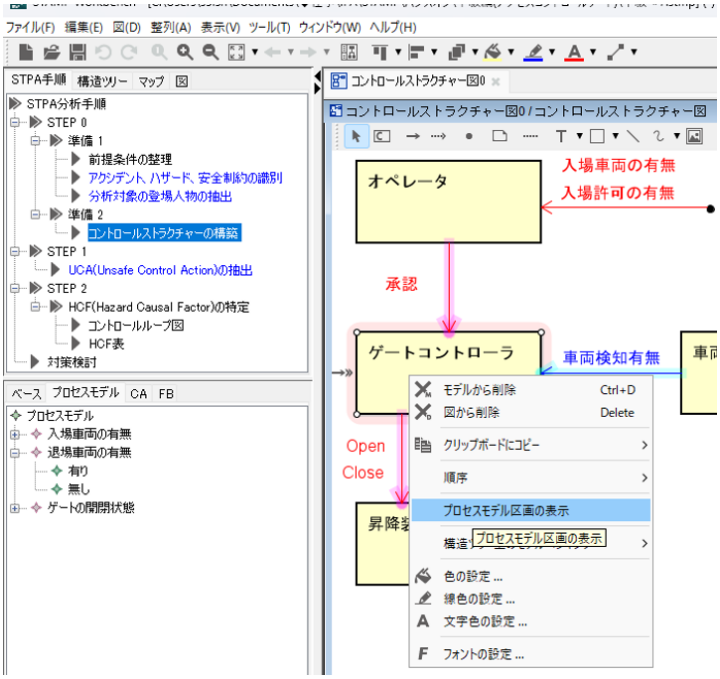


3. プロセスモデルの入力もこの画面のプロセスモデルタグを押してから入力、編集できる。

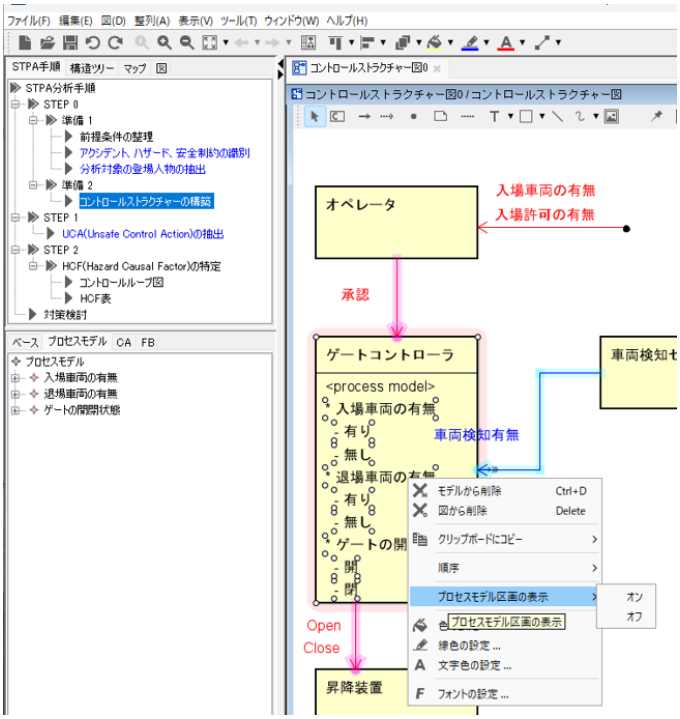


プロセスモデルを表示する

プロセスモデルをコンポーネント内に表示するには、CS 図でコンポーネントを選択し、コンテキストメニューを開き（右クリック）、「プロセスモデル区画の表示」を選択する。



プロセスモデルをコンポーネント内に表示すると、表示する分だけコンポーネントが大きくなるので、CS 図内でのコンポーネント配置を再度編集する必要がある。プロセスモデルを表示してみて見難いと思ったら、「プロセスモデル区画の表示」をオフにして戻せば良い。



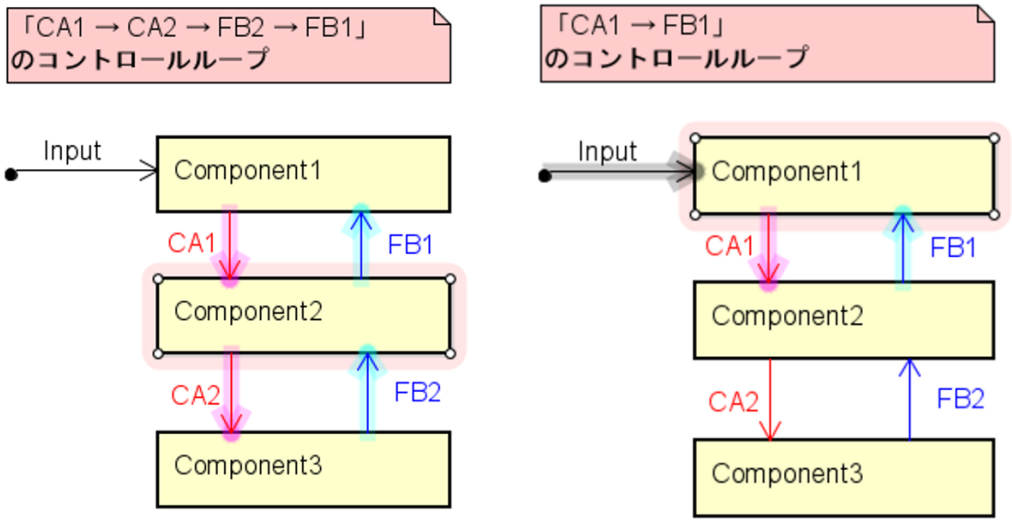
分析している最中に、一時的にプロセスモデルを表示したいと思った時だけ表示するのでも良い。UCA を識別するときにはプロセスモデルの表示がとても有用だが、分析結果をレビューするときには相互作用に注目して欲しいのでプロセスモデルの表示は目障り、というケースもある。

Power Point では難しいことですが、STAMP Workbench はモデリングツールなので、表示のオン/オフ切り替えは容易ですし、切り替えの際に修正ミスも有り得ません。利用場面に応じて柔軟に切り替えてご利用ください。

■STAMP 及びツールのノウハウ

UCA はコントロールループで考える

UCA を識別するときには、CA と FB から成るコントロールループの単位を対象として考える。
下図のような CS 図において、外界からの input をトリガーとして、必ず「CA1 → CA2 → FB2 → FB1」の順番となるシステムもあります。その場合、CA1 に関する UCA を識別するのに「CA1 → FB1」のコントロールループで考えるのか、「CA1 → CA2 → FB2 → FB1」のコントロールループで考えるのか、悩ましいことがあります。
結論から言うと、UCA の識別においてはどちらでも構いません。
但し、ハザードシナリオを抽出する際には、「CA1 → CA2 → FB2 → FB1」のコントロールループの方が考え易いかもしれません。
UCA の識別では「CA1 → FB1」のコントロールループで考えて、ハザードシナリオの抽出では「CA1 → CA2 → FB2 → FB1」のコントロールループで考える、というのでも構いません。
STAMP/STPA は柔軟な手法であって、このような細かなルールを強制しません。



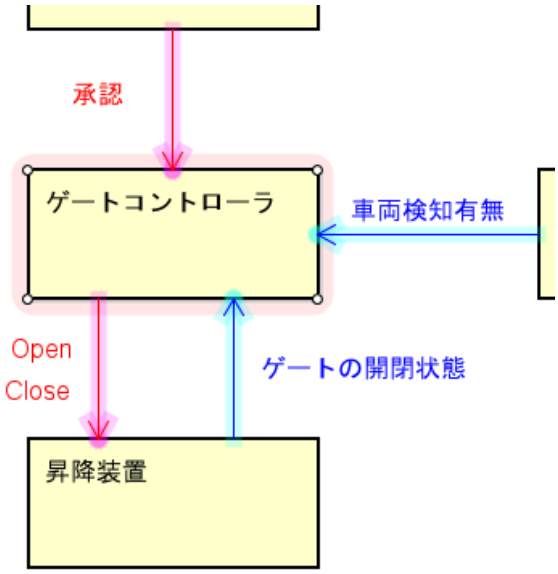
UCA の識別では原因を考えない（思いついても我慢）

UCA を識別するときには原因を考えないようにします。原因を考えながら UCA を識別しようとすると、原因を思いつかないケースでは UCA が識別できなくなります。原因を先に考えたら、対象システムについての知見が有るか否かに分析結果が依存することになります。
UCA の識別では、4 つのガイドワードを参考にして、漏れの無い『観点』から論理的に識別します。STAMP/STPA において、『観点』として漏れが無いことを示せることが重要で、分析の有効性を説明する際の肝となります。
それでも原因を思いつくことはあります。それは不必要な思いつきではなく、次のステップで HCF やハザードシナリオを特定する際にとても有用な思いつきです。その貴重な思いつきは捨てることなくメモに残しておき、次のステップに役立てます。
大事なことは、UCA を識別するときには先に原因を考えない。思いついてもその原因だけに捕らわれないように我慢することです。

コントロールループ図は必要？

コントロールループ図（CL 図、CLD）は必ずしも描かなくても良いです。
コントロールループ図を描く目的は、CA・FB からなるコントロールループの中に存在し得るハザード要因（HCF）を探しやすくすることです。CS 図でコンポーネントの数が多く、少し複雑な場合には着目する CA のコントロールループが見づらくなるため、対象コントロールループだけを抜き出してコントロールループ図を作成します。しかし、それほど複雑な CS 図でなかったり、コントロールループを見失うことが無い CS 図ならば、コントロールループ図を作成する意味がありません。

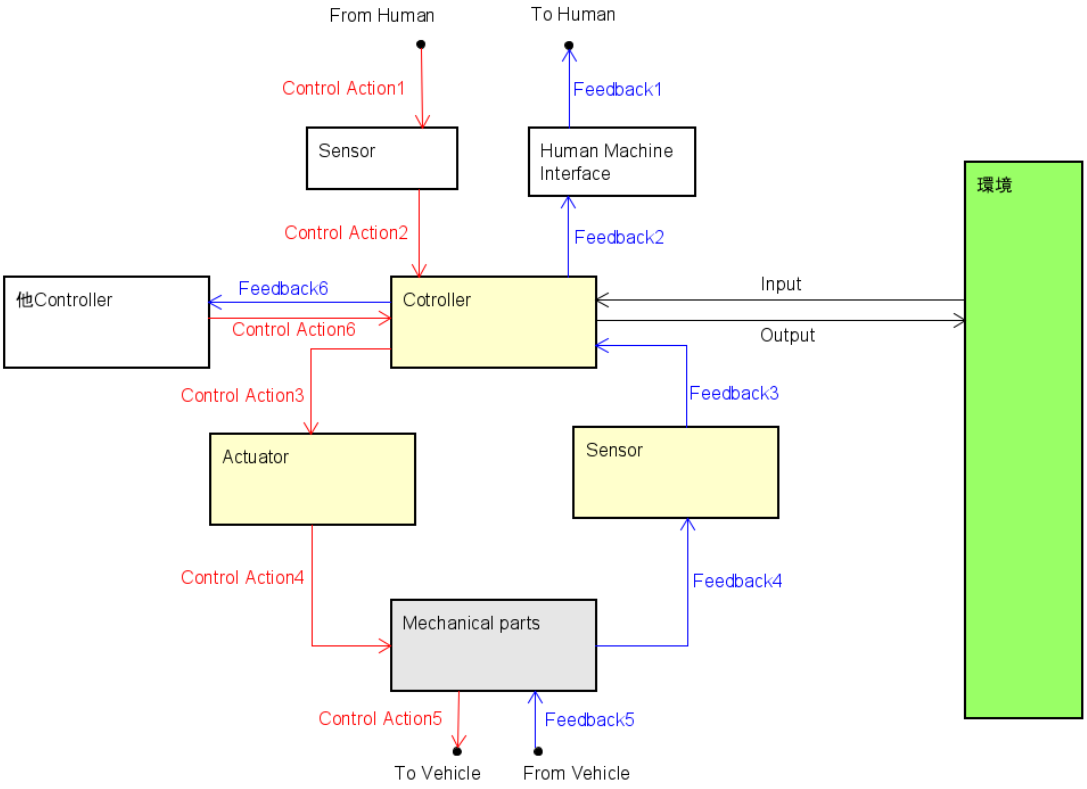
下図では、Open/Close の CA を含むコントロールループは、Open/Close の CA とゲート開閉状態の FB から成ることが明確なので CL 図を作成するまでもない。下図のように対象 CA を発出するコンポーネントを選択すると、そのコンポーネントが直接関与する CA 線と FB 線がハイライトされて見易くなる。



外界モデル

コントロールストラクチャー図（CS 図、CSD）やコントロールループ図（CL 図、CLD）において、一般的にはシステム外のコンポーネントは記述しません。しかし、外界の何とのインタラクション（何から input を受け取る／何に対して output を発出する）かを記述しておきたい場合もあります。その場合は、外界モデルとしてコンポーネントを記述する方法が都合が良いでしょう。
但し、気を付けて欲しいことは、外界モデルとの CA や FB のやり取りは無いはずです。外界モデルとのやり取りは input/output だけになります。CA や FB が出てきたときにはシステム境界の定義が曖昧になっていると考えられるので、現在の分析対象範囲はどこまでなのかを再確認してください。

下図では分析において「環境」を意識するように外界モデルとして明記。



STAMP 用語

STAMP 解説書では次の用語を定義している。

Accident：事故（アクシデント）。 損失（人命の損失や負傷、経済的損害、環境汚染などを含む）をもたらす、望ましくない、計画されていない事象。
Hazard：ハザード。 最悪の環境条件において事故（損失）（accident (loss)）につながるシステムの状態（system state）または条件の集合（set of conditions）。または潜在危険。
※Engineering a Safer World 翻訳本「システム理論による安全工学」より

これらは、一般用語としての用語定義ではなく、STAMP 用語としてのローカル定義なので、一般用語としての用語定義と勘違いしないこと。英語辞書にも日本語辞書にも accident やアクシデントに損失という意味は含まれない（辞書によっては、特殊な例で、そういう記述もあるかもしれないが）。
アクシデントを損失と定義するのも、ハザードをシステム状態と定義するのも、STAMP を広く適用できるようにするための工夫と考えられる（本 Tips 筆者の個人的な推測である）。
例えば、「アクシデント」は、人命の損失や身体の損傷だけでなく、所有物の毀損、経済的損失、ミッションの未達なども該当する、と解説している。アクシデントを一般用語の通りに「事故」と捉えるとミッションの未達等はアクシデントに含め難いが、アクシデントを「損失」と捉えることによって、その解説が尤もらしくなる。

STPA 手順は変化している？

結論から言うと、STPA の手順は変わっていません。

皆様が良く参照される STPA 手順解説書には次の 3 つがあると思います。

(1) 2012 年 MIT 発行の「STPA Primer」

公開終了。Engineering a Safer World における解説と同じ。

(2) 2016 年 IPA 発行の「はじめての STAMP/STPA」

https://www.ipa.go.jp/digital/stamp/about.html

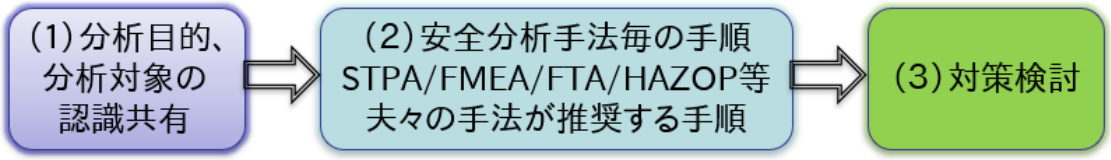
(3) 2018 年 MIT 発行の「STPA HANDBOOK」

http://psas.scripts.mit.edu/home/get_file2.php?name=STPA_Handbook_Japanese.pdf

これらの解説書に記述された STPA 手順の解説が異なるので、STPA 手順が変化していると感じられるかもしれませんが、それぞれの解説範囲と表現が異なるだけで、STPA 手順の基本は同じです。変わっていません。

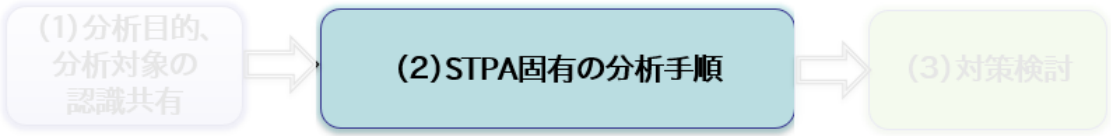
それぞれ理由・背景があって表現や解説範囲を変えています。その理由・背景は汲み取っていただきたいところですが、「STPA 手順が変わった」などと過度に反応されることのないようお気を付けください。

【一般的な安全分析手順】



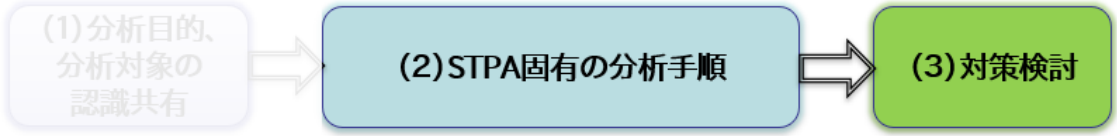
活用する安全分析手法が何か（STPA か、FMEA か、FTA か、HAZOP か等）に依らず、(1)⇒(2)⇒(3)が一般的な安全分析の手順である。

【STPA Primer】



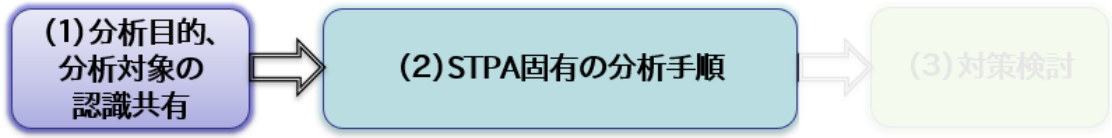
純粋に STPA 固有の手順についてのみ解説している。

【はじめての STAMP】



対策検討のステップを含めて解説している。IPA の STAMP 向けモデリングツール STAMP Workbench は、この表現を用いている。

【STPA HANDBOOK】



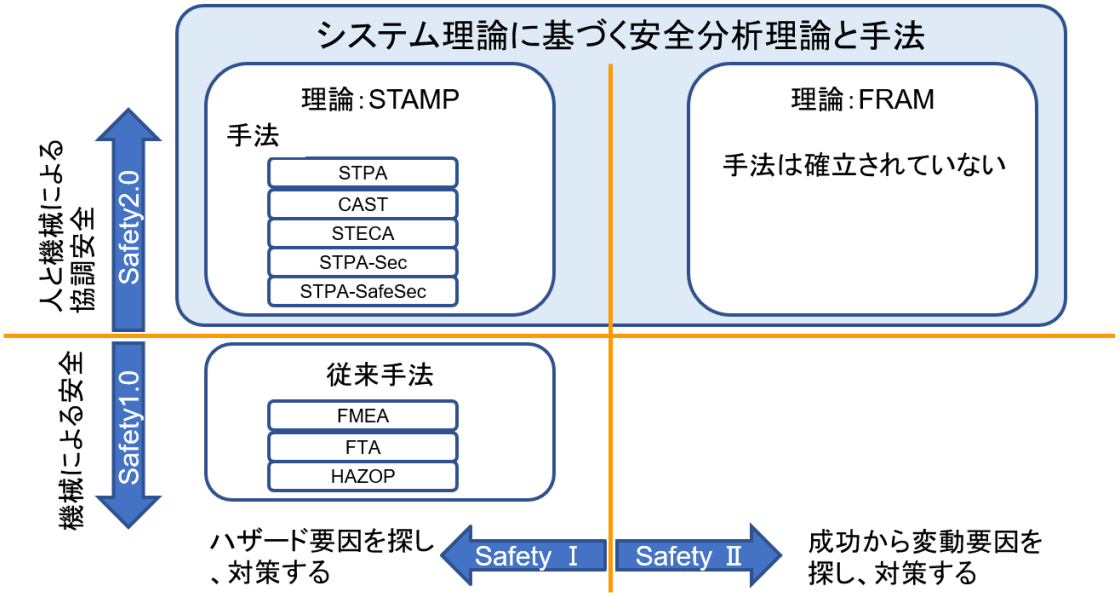
「STPA Primer」に記された STPA 手順に分析目的・分析対象の整理・認識共有の重要性を解説に加えた。

表現と解説範囲に差異があるが、いずれも(2)の STPA 固有の分析手順に差異はない。

STAMP と FRAM、Safety-Ⅱと Safety2.0

IoT 時代の複雑システムに対するこれからの安全分析では、**システム理論**に基づき、システムを俯瞰して分析する必要性が増している。また、止める安全（Safety1.0：機械による安全）から止めない安全（Safety2.0：**人と機械による協調安全**）への要望が増している。そして、ハザード要因から守る（Safety-Ⅰ）だけではなく、**成功から学ぶ**（Safety-Ⅱ）という安全戦略も注目されている。
このような背景においてシステム安全に関わる理論や手法として STAMP(その代表的な分析手法が STPA)や Resilience Engineering（その代表的なモデリング手法が FRAM）が提唱されている。

STAMP と FRAM、Safety-Ⅱと Safety2.0 について、その違い・関係性を俯瞰するシンプルな分類および分類図をここに紹介する。



Copyright © 2018 IPA, All Rights Reserved

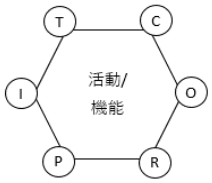
※ STAMP と FRAM、Safety-Ⅱと Safety2.0 に関し、この 4 象限で示す分類と分類図は、多少乱暴ではあるが関係を俯瞰するのに役立つと思い、筆者が IPA にて作成したオリジナルの分類と分類図である。この分類および分類図も「CC BY」のライセンスのもとに使用を許諾する。

FRAM を応用した STPA 手法

FRAM 分析を実施するには Resilience Engineering に関するかなり深い知識と独特な感性が要求されるが、ここでは STPA において「Resilience 性についても意識する」という程度の FRAM 応用手法（言うなれば“FRAM 風味の STPA”）を紹介する。

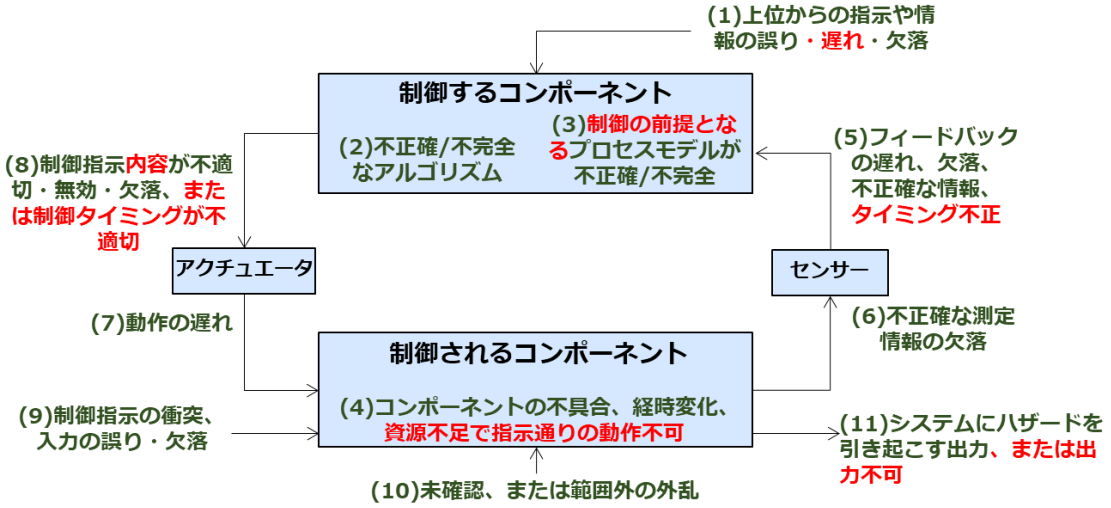
下図は、Resilience Engineering における機能間の関係を図示するモデリング手法 FRAM での、機能を表現する要素である。

I,T,C,P,R,O の 6 つの側面（aspect　視点）で表現している。



I(入力)	Input	機能が処理を行い、変化するもの。あるいは、機能を開始するもの。
O(出力)	Output	機能の結果、何らかの実態か状態変化。
T(時間)	Time	機能に影響を与える時間制約(開始時刻、終了時刻、継続時間に関して）。
C(制御)	Control	機能がどのようにモニターされる、あるいは制御されるか。
P(前提条件)	Preconditions	機能が実行される前に存在すべき条件。
R(資源)	Resources	機能が実行されるときに必要とされるもの(実行条件)、あるいは、出力を提供するために消費されるもの。

STAMP/STPA の HCF（ハザード誘発要因）を特定する際のヒントワードに FRAM の 6 つの視点を取り込んだ。



(1) コントロール入力や外部情報の誤り、遅れや喪失

(2) 不適切なコントロールアルゴリズム(作成時の欠陥、プロセスの変更、誤った修正や適用)。適切な制御判断不可

(3) 前提となる制御出力条件が揃わず、不整合、不完全、または不正確なプロセスモデル。不適切な操作、不適切な制御判断

(4) コンポーネントの不具合。経年による変化。必要な資源不備で指示通りの動作不可

(5) 不適切なフィードバック、あるいはフィードバックの喪失。フィードバックのタイミングが早過ぎる／遅すぎる

(6) 不正確な情報の供給、または情報の欠如。測定の不正確性。フィードバックの遅れ

(7) 操作の遅れ

(8) 不適切または無効な内容のコントロールアクション、コントロールアクションの喪失。制御タイミングが不適切

(9) コントロールアクションの衝突。プロセス入力の喪失または誤り

(10) 未確認、または範囲外の障害

(11) 出力能力が不足し出力できない、システムにハザードを引き起こすプロセス出力

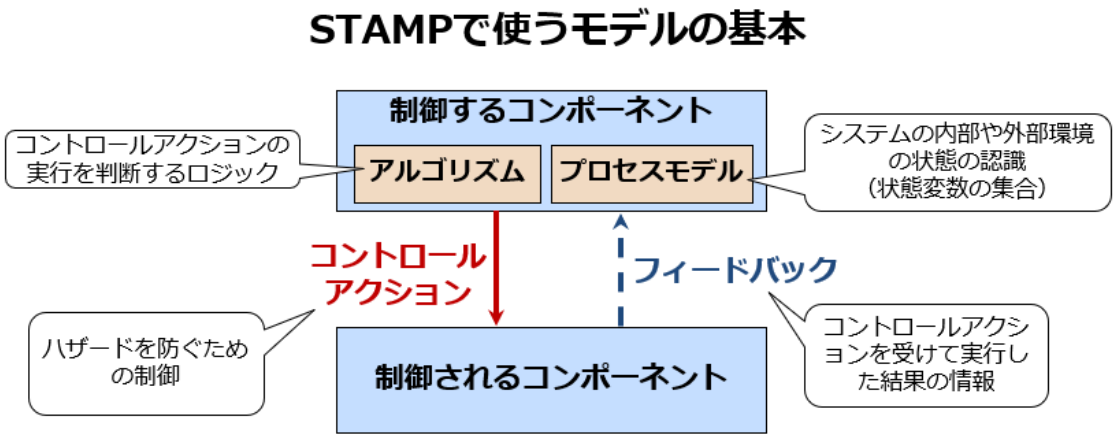
基のヒントワードの(1)～(11)と、FRAM の 6 つの視点 I,T,C,P,R,O の対応関係を以下に記す。

FRAM の視点	視点の意味	出力に与える影響 ハザード要因としての読み替え	対応するヒントワードの番号
I：	機能のトリガー	即時に処理が開始され、出力される	(1), (7),(5)

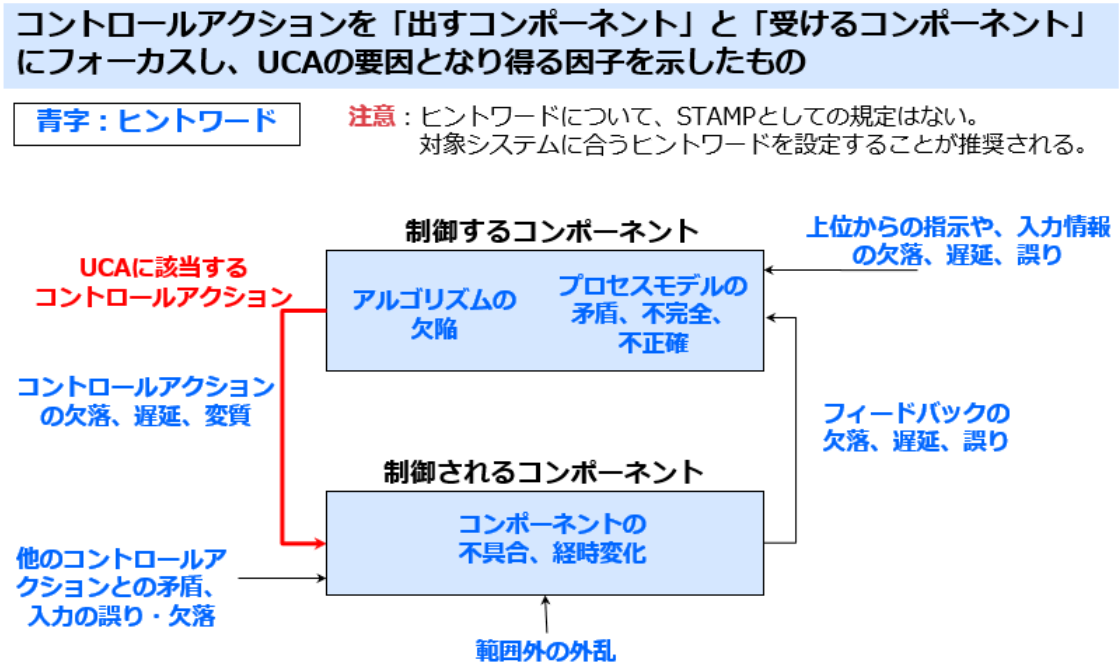
入力		トリガーとなる入力の遅れ、欠落により出力が遅延、欠落	
P： 前提	機能の実行に必要な前提条件	前提条件として常にIの前に入力されている必要有 前提入力不足・誤りで、プロセスモデルが不正確／不完全のため誤指示、指示欠落	(3)
R： 資源	機能の実行に必要な資源	暫時減少するため、枯渇により出力は停止する 出力に必要な資源枯渇・不足により出力不可	[(1),] (4),(11)
T： 時間	機能の実行における時間制約	時間制約により出力を許可･禁止する（非線形的効果） 誤ったタイミングで制御指示を発出	(3),(5) ,(8)
C: 制御	機能の実行を制御するパラメータ	出力値を直接変動させる 正しい制御指示を生成できない。誤指示、範囲外指示	(2),(3) ,(8)
O： 出力	機能の出力		

制御される側（被制御コンポーネント）にもプロセスモデルがある

STAMP で使用するモデルの基本要素は下図のように制御する側（制御コンポーネント）と制御される側（被制御コンポーネント）とそれらの間に相互作用である制御（CA)、フィードバック（FB）が有り、制御する側は制御アルゴリズムと制御内容／可否判断材料となるプロセスモデルで構成される。

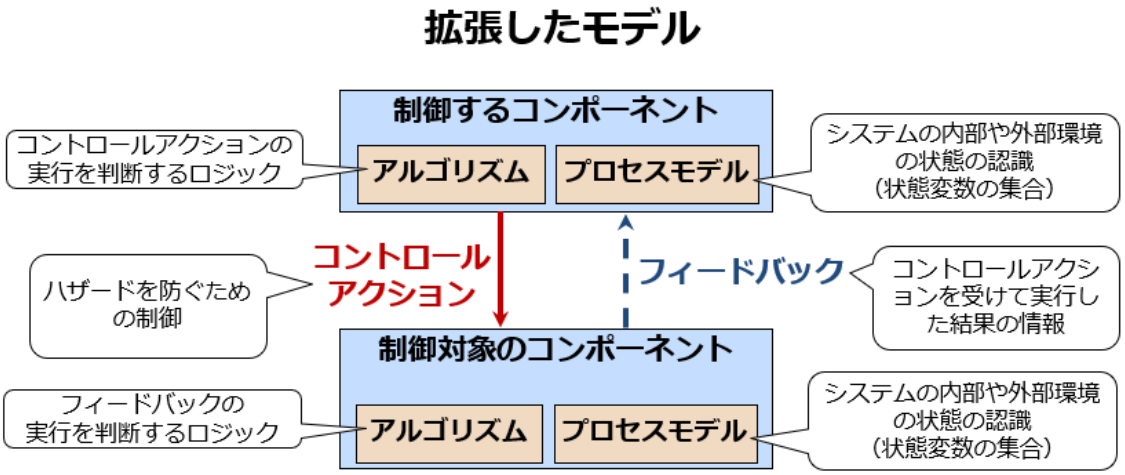


上記モデルに関するハザード誘発要因（HCF）を考える際のヒントワード

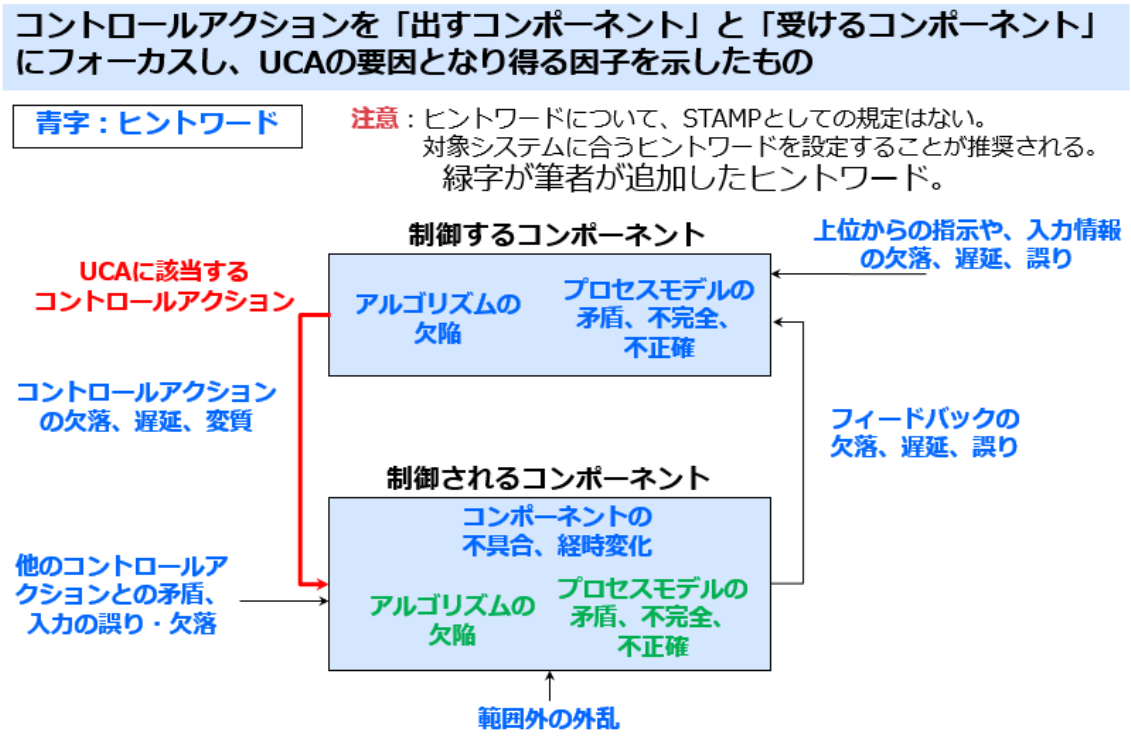


一般的には上記の通りであるが、場合によっては制御される側（被制御コンポーネント）にもプロセスモデルが存在することを意識すべきことがある。特に、人と人、組織と組織、人と組織から成るシステムにおいては、ハザード誘発要因（HCF）やハザードシナリオ（ロスシナリオとも呼ぶ）を考えるときに制御される側のプロセスモデルを意識すべきケースが少なくない。

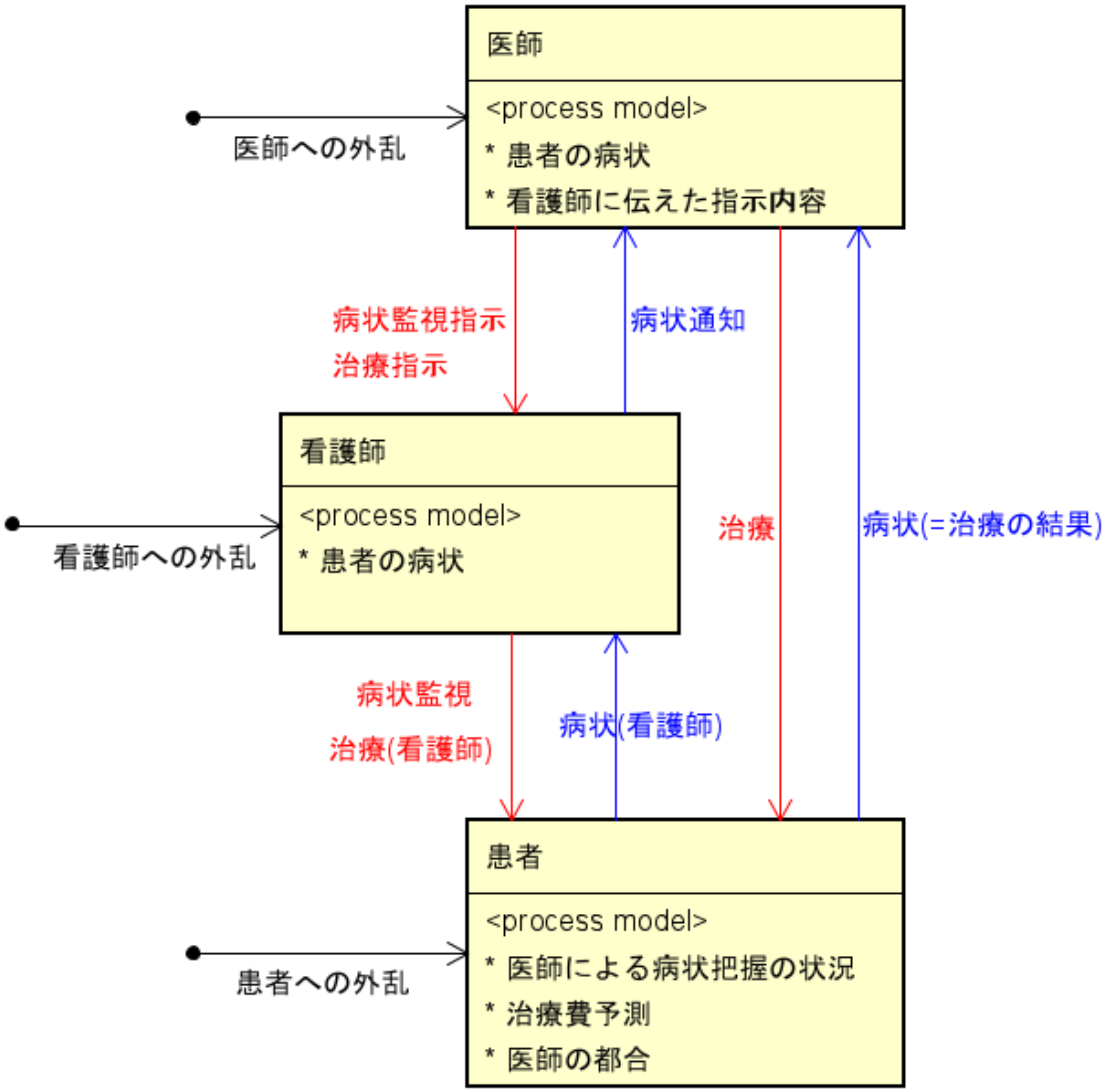
下図は被制御コンポーネントにもプロセスモデルを設定して拡張したSTAMPモデルの基本要素である。



上記モデルに関するハザード誘発要因（HCF）を考える際のヒントワードをここに提案するので、使えるケースがあれば活用いただきたい。



下図は医療における人と人のコミュニケーションエラーに着目した CS 図の例で、制御される側のプロセスモデルを意識しなければ、ハザード誘発要因に気付かない可能性が高い。



尚、この「制御される側にもプロセスモデルが有る」という考え方は、STAMP の原著 Engineering a Safer World や、STAMP/STPA 解説書の STPA HANDBOOK には記述が無いので、STPA の派生手法、あるいは STPA の発展手法とも言えるものだが、ハザード誘発要因を検討する際にとても役に立つ手法と思われるので、ここで紹介する。お試しいただきたい。
また、この手法は、人対人、人対組織、組織対組織のシステムだけでなく、人とコンピューター、人と機械、コンピューターと機械などあらゆる組み合わせのシステムにおいても通ずるものと、筆者は考えている。

※　制御する側と制御される側の双方のコンポーネントにプロセスモデルが存在することは、早稲田大学理工学術院 人間生活工学研究室の小松原明哲教授が提唱された。小松原先生は、どの相互作用に着目するかによって、制御する側と制御される側の主従を入れ替えたモデル（CS 図）を使って分析することの有効性を述べられている。
この小松原先生の提案にヒントを得て、コンポーネントの主従を入れ替えることなく、FB に対するアルゴリズムとプロセスモデルの存在を仮定し、改訂版ヒントワードを用いたハザード誘発要因（HCF）検討の手法を筆者が提案する。この提案は小松原先生が提唱される「制御される側にもプロセスモデルが存在する」ことを前提とし、小松原先生の提案の見る向きを変えたものと言える。
小松原先生の提案について詳しくは「患者安全推進ジャーナル 77 号」p.41-46 特別寄稿を参照ください。

https://www.psp-jq.jcqhc.or.jp/post/journal/18835

双方向の CA が出てくる

2 つのコンポーネント間で双方向の CA（Control Action）が出てくることがある。

結論）異なるハザードについて同時に考えているからだと思われる。

説明）分析対象が同じシステムでも、分析対象のハザードが異なれば CS 図が異なる場合がある。双方向の CA をひとつの CS 図を用いて同時に分析するのではなく、制御する側と制御される側の主従を入れ替えた CS 図も用いて、それぞれの CA を別の CS 図で分析する方法がある。

例）

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

Controller（制御する側）と Controlled process（制御される側）の関係は、 分析者がどの関係性（相互作用）をどちらから見たいのか（何を見たいのか）によって異なる。例えば、

政治が調子良い時：

Controller＝政治家

Controlled process＝有権者

政治が調子悪い時や選挙のとき：

Controller＝有権者

Controlled process＝政治家

この二つは同じシステムを扱っていても、分析対象のハザードが異なるから CA が逆方向になっている。どちらも間違いではない。

今、分析したいハザードがどちらなのかを再度、ステークホルダーと議論して認識を共有しなおすと良い。 両方のハザードを同時に分析すると混乱する可能性がある。

※ 本 Tips ページの「**制御される側（被制御コンポーネント）にもプロセスモデルがある**」も参考にして欲しい。

対策検討の対象は HCF かシナリオか？

シナリオ（ハザードシナリオ、ロスシナリオとも呼ぶ）に対して対策を検討するのか？ それとも HCF（ハザード誘発要因）に対して対策を検討するのか？

HCF に対して対策を検討すべき。

シナリオを睨みながら考えると対策を思いつき易い。また、シナリオはテストシナリオにもなるので、とても重宝する。よって、安全分析でシナリオを特定しておくことは重要である。

但し、対策が特定のシナリオに特化したものにならないように注意すべき。特定のシナリオに特化した対

策では、同じ HCF の別シナリオに対応できない対策になってしまう恐れがある。
対策検討時には、特定のシナリオのみに縛られないようにする。

HCF に対して対策を考えたら、その対策がすべてのシナリオの対策になっていることを確認することで、対策の十分性を確認できる。

シナリオを特定することのメリットとして次のことが考えられる。
- 対策を思い浮かべ易い。
- 対策の十分性を論理的に検証し易い。
- 対象 HCF が発生し得るものであることを示せる。
- コンポーネント開発者へのアドバイスになる。
- テストシナリオになる。
- HCF を説明し易く、認識共有し易い。

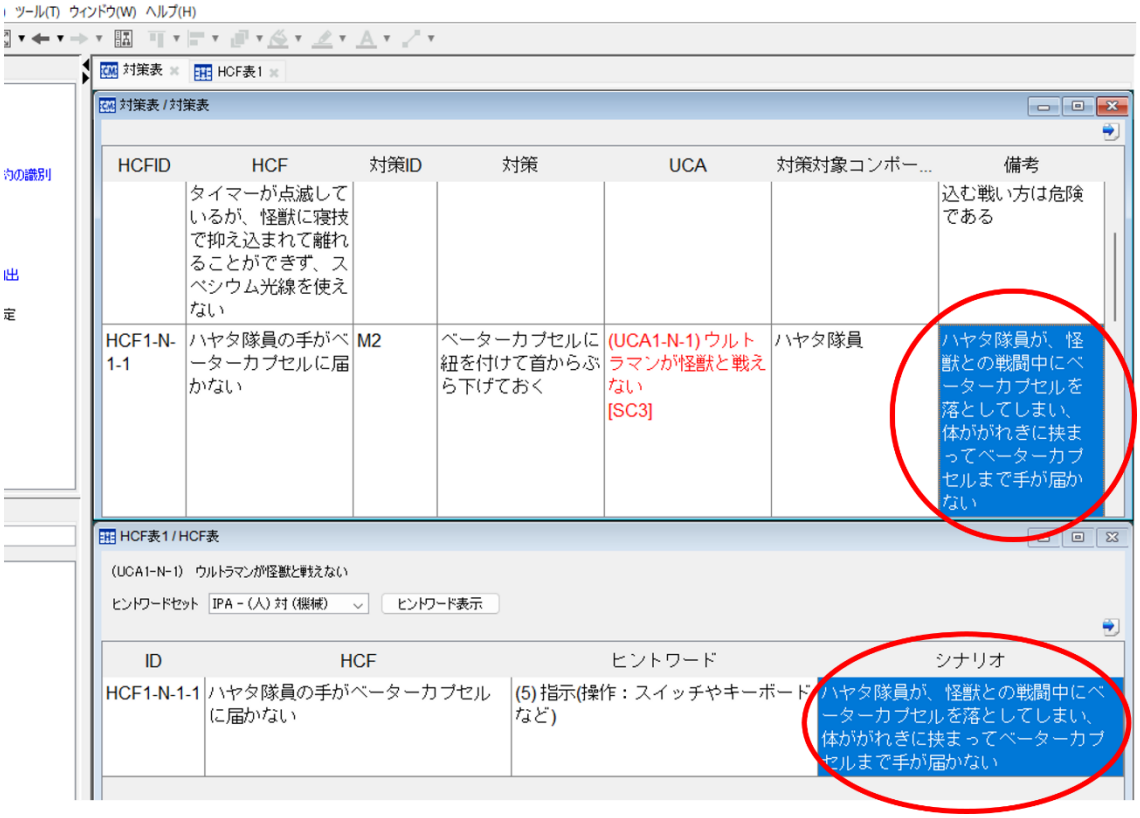
決して、シナリオを作成しなくても良いということではないので、勘違いしないで欲しい。HCF を特定したら、その HCF を含むシナリオを考え、作成することは上記のようにとても有用なので、是非シナリオ作成して欲しい。
シナリオをたくさん思いつくならば多いに越したことはない。しかし、発生する可能性のある全てのシナリオを網羅することなど不可能である。極端に言えば、シナリオを一つ特定できて、対象 HCF が発生し得るものであることを示せれば良い。一方、シナリオが多ければ多いほど、多くの対策案を思いつき易く、対策の十分性も確かめ易くなるので、シナリオ作成を軽視しないこと。

対策表の備考欄にシナリオをコピーする

対策表の備考欄に、手動で HCF のシナリオをコピーしている。

ツールへの要望
このコピー作業を自動化するか、対策表に列を追加してシナリオが自動的に表示されると良い。
この対策はツールが自動的にサポートすることが望ましいが現状サポートできていません。

下図では、HCF 表と対策表の 2 つの Window を上下に並べて 2 画面表示しているが、対策表の備考欄にシナリオをコピーしておけば 2 画面表示しなくても済む。



対策表に【機能】【運用】【教育】など分類を記載した

対策のところに【機能】【運用】【教育】など分類を記載し後から見やすくした。
対策表を Excel 出力して、列を増やし、それぞれの対策に【機能】【運用】【教育】などの分類を記載して、後で他の人が対策表を見たときに、次の開発ステージであるコンポーネント設計に対するコンポーネント安全制約なのか、運用で考慮するという対策なのか、設計者・運用担当者に対して教育すべき事項なのか、が分かるようにした。

このように自社独自、自プロジェクト独自の ID や分類などを表に追加する機能は STAMP workbench に備わっていないので、Excel ファイルに出力してから Excel で列を追加して対応した。

人のコミュニケーションの対策例 3way コミュニケーション

人のコミュニケーションの対策例として 3way コミュニケーションという方法がある。

3way コミュニケーションとは伝達→復唱→確認のこと。
(「指示する」、「指示を復唱する」、「指示をもう 1 回繰り返して確認する」)

伝達：「○○してください」
復唱：「○○ですね？」
確認：「はい、○○です」

原子力業界では一つの操作ミスが重大なトラブルに繋がったり、地域住民の信頼を損ねることに成り得るためコミュニケーションは重要な課題として捉えられており、3way コミュニケーションが行われてい

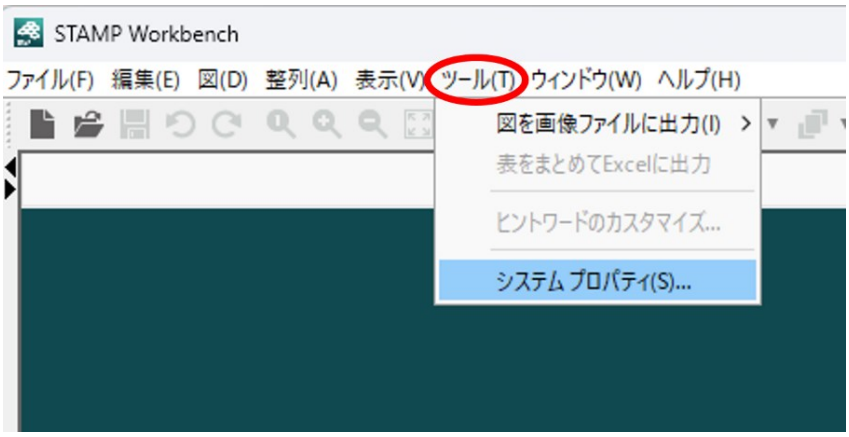
るそうだが、3way コミュニケーションの伝達→復唱→確認という一連の流れは原子力業界以外でも重要な指示に対して有効と考えられる。対策例として参考にして欲しい。

■注意事項

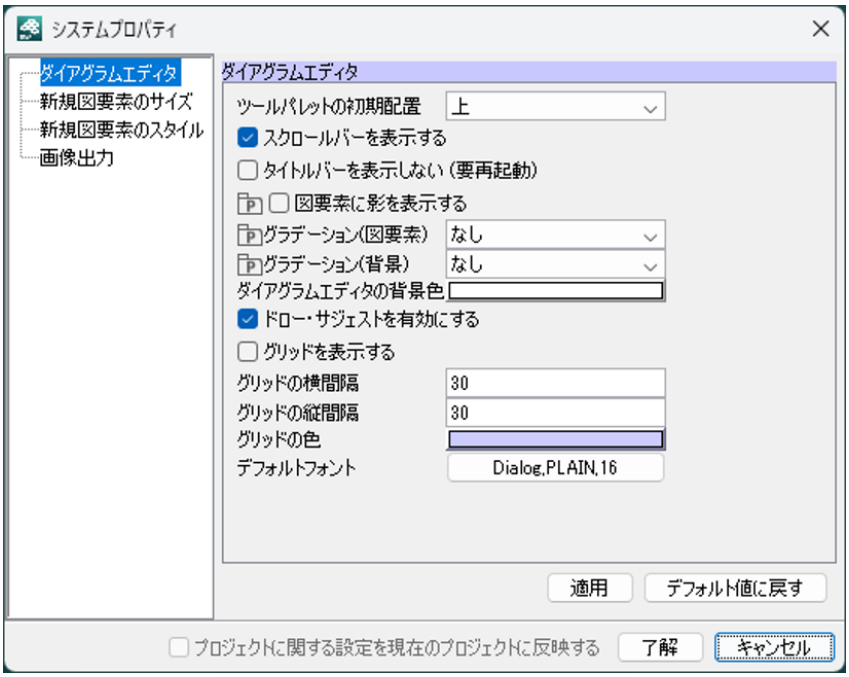
ダークモード（ハイコントラストモード）

Windows の設定で「ハイコントラストモード」を有効にして、「ハイコントラスト：黒」とすると一部の文字が読めない問題が確認されています。STAMP Workbench は「ハイコントラストモード」での文字・図形色自動変更に対応していないため、黒地に黒文字／黒線のような設定となり文字が読めなくなる現象です。ツール -> システムプロパティの「ダイアグラムエディタ」や「新規図要素のスタイル」で背景・図形・線・文字の色を選択することをお試しください。

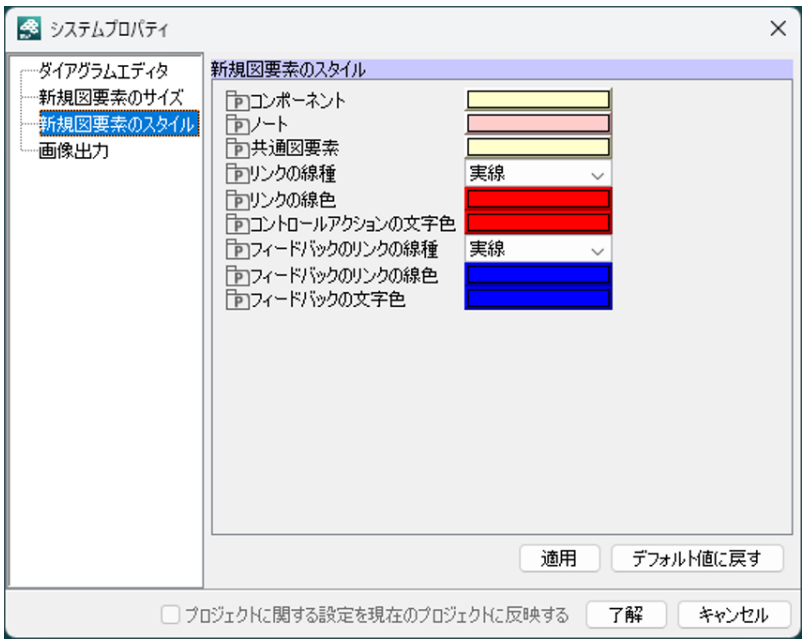
1. システムプロパティ画面を起動



2. 「ダイアグラムエディタ」で背景や図要素の色を変更



3. 「新規図要素のスタイル」で文字・線色を変更



Excel 出力時の問題

MS Office の性能限界への対応方法です。Excel では既定の列幅最大値が 255 文字（MS Office のバージョンに依る）となっているため、STAMP Workbench で表の列幅（セル幅）を 256 文字以上にした場合、Excel 出力時にエラーが発生する可能性があります。
4K 以上のディスプレイで高解像度設定し、かつ縮小表示して、STAMP Workbench のウィンドウを全画面表示し、更にセル幅が 255 文字を超えるようにした場合、STAMP Workbench の表を Excel 出力したときにエラーが発生します。但し、エラー発生時に STAMP Workbench が保持するデータが壊れたり、失われたりすることはありません。
Excel 出力を実施する前に、STAMP Workbench のウィンドウ幅を少し小さめに変更することで問題を回避できます。

バージョンアップ時の注意事項（DL ページの再掲）

既に STAMP Workbench をご利用の方がバージョンアップされる場合は、旧 version を一旦アンインストールまたは削除してから新 version をインストールしてください。

実行形式インストーラーで旧 version をインストールされた方

インストール済みの旧 version を一旦アンインストールした後に、最新の実行形式インストーラー（msi ファイル）を実行してご利用ください。

旧 version の zip 版をインストールされた方

zip を展開して利用していた旧 version のフォルダーを削除した後に、最新の MSI 形式インストーラーを実行するか、最新の zip ファイルを展開してご利用ください。

旧 version をアンインストールあるいはフォルダーを削除しても、旧 version で行った設定内容は新 version に自動的に引き継がれます。

画面表示に問題がある場合

新 version の STAMP Workbench をインストールした後、一部の文字が読めなくなるなどの問題が確認されています。最新 version をインストールした際、古い version のファイルが残っていると、この画面表示の問題が発生する場合があります。古い version をアンインストールしたときにアンインストーラーでは古いファイルを削除しきれない場合があるためです。以前にアンインストールせずに上書きインストールしたことがある場合に、このような状態になります。この問題が発生した場合には、下記手順をお試しください。

1. STAMP Workbench を終了する
2. アンインストーラーでアンインストールする
3. 元インストールしていた stampworkbench のフォルダーを完全に削除する（デフォルト設定では、C:¥Program Files¥stampworkbench フォルダー）
4. 再度 STAMP Workbench をインストールする