システム思考のモデリングはこれからのソフトウェアプロセスに有効か？

日下部 茂
長崎県立大学
kusakabe@sun.ac.jp

岡本 圭史
仙台高等専門学校
okamoto@sendai-nct.ac.jp

要旨

ソフトウェアの多様な利活用が進み，ソフトウェアのライフサイクルにおいて実世界や様々なシステムとのつながりを考慮することが必要となっている．筆者らはシステム思考のモデリングはこれからのソフトウェアプロセスに有効と考え，研究レベルのいくつかの取り組みを行っている．本フューチャープレゼンテーションでは，これまでの取り組みを紹介し，システム思考のモデリングはこれからのソフトウェアプロセスに本当に有効かの議論を行いたい．

1. はじめに

ソフトウェアは，計算だけに限らず，多様な目的での利活用が進み，そのライフサイクルにおいて，実世界や様々なシステムとのつながりを考慮することが必要となっている．安全性やセキュリティといった，様々な相互作用の分析を必要とする特性の重要性が高まっており，システム思考のモデリングはこれからのプロセスに有効と考えている．システムズエンジニアリングとそこで必要とされる分析を含むプロセスの関係はこれまでも議論も行われており，システム思考に関しても様々な枠組みが提案されている．ここではシステム思考の枠組みにはソフトウェア集約システムに有効とされている STAMP/STPA[3]を，プロセスについては例として ESPR をレファレンスとして議論を行う．有効性検討の事例として，アーキテクチャ記述言語 AADL[1]で記述したモデルを用いて STAMP/STPA, Fault Impact Analysis (FIA), Fault Tree Analysis (FTA)を実施して比較した例を紹介する．

2. ソフトウェア集約型システムのモデリング

STAMP は従来の解析的還元論や信頼性理論ではなくシステム理論に基づくハザードのモデル化と分析のために提唱されたもので，コンポーネント間の相互作用によって生じる創発的なものを含め，対象システムのハザードをコントロールするという観点に着目した事故モデルである(図 1 参照).

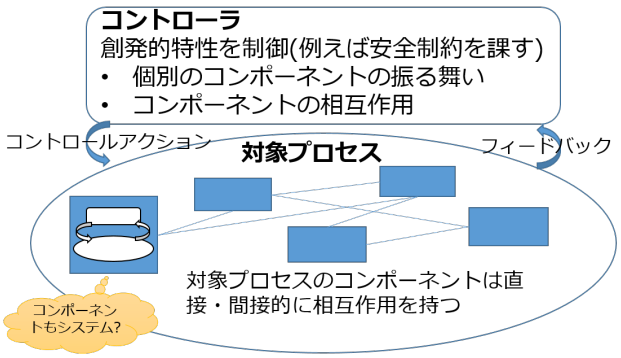

図1 対象プロセスに対するコントロール

STAMP/STPA はソフトウェア集約型のシステムが登場する以前に提唱された FTA や FMEA と異なり，ソフトウェア集約型システムを想定したものであるため，ソフトウェアのプロセスで効果的なモデル化と分析が可能と考える．

STAMP のモデルは，安全制約，階層的なコントロールストラクチャ，プロセスモデルという三つの基本要素で構成されており，コントロールストラクチャとプロセスモデルに対して，システムの安全制約が正しく適用されているかどうかに着目する．ここでコントロールストラクチャは，システムを制御する各機能の相互作用の構造を表すもので，コンポーネント間でやり取りされる制御の指示やフィードバックなどを表す．STAMP は安全工学の知見を広く活用することを念頭に提唱されており，モデリングの際の「事故」を「受容できない損失の発生」ととらえると，ソフトウェア開発での様々な関心事に適用可能とされている．

3. システム開発プロセスでの活用例

例えば組込みシステム開発のリファレンスプロセス ESPR[2]で考えると，SAP1：安全要求仕様書の作成や，

SYP2：システム・アーキテクチャ設計，SWP2:ソフトウェア・アーキテクチャ設計など FMEA や FTA といったハザード分析の実施が想定される部分では STAMP/STPA を活用可能と考える．

そのようなプロセスでの活用を念頭に，アーキテクチャ記述言語 AADL[1]で記述したモデルを用いて STAMP/STPA，Fault Impact Analysis (FIA)，Fault Tree Analysis (FTA)を実施した事例を紹介する．この事例の実施目的は，STAMP/STPA と他 2 手法の分析結果比較であり，特に STAMP/STPA の分析結果が他手法で得られるかを調査することを目的とした．

3.1. STAMP/STPA と AADL

STAMP/STPA によるハザード分析では，システム全体の振る舞いを示すモデルである，コントロールストラクチャを作成する．そして各コンポーネント間の相互作用，コントロールアクションの識別する．次にコントロールアクションの中からハザードを引き起こすものを非安全制御動作(Unsafe Control Action, UCA)として識別する．そして UCA がどのようにハザードを引き起こすのかの経緯を，必要に応じて更に詳細なモデルを作成するなどして分析し，誘発要因（Causal Factor）を見つける[3]．

システム開発プロセスの中では，仕様からシステムの構造や振る舞いをモデル化する．他方，STAMP/STPA ではハザード分析を行うためにシステムのモデルとしてコントロールストラクチャを構築する．これらの二種類のモデルを統合したモデルとして，アーキテクチャ記述言語 AADL とその安全分析用拡張である Error Model Annex を用いたモデルが提案されている[4]．

3.2. 分析

初めに，学習用に公開されている要求仕様を基に，AADL モデルを記述した．次に，作成した AADL モデルの抽象度を整理したものを STAMP/STPA のコントロールストラクチャ図とし，それを基に STAMP/STPA を実施した．最後に，この AADL モデルに，FIA に必要なエラー伝搬と，FTA に必要な状態遷移を Error Model Annex を用いて追記し，ツールによる FIA と FTA を実施した．なお，FIA，FTA に必要な情報を網羅的に記述し，分析を実施するのはコストがかかるため，STAMP/STPA で得られたハザードシナリオを選別しそのハザードシナリオに関連する情報のみを追記し，分析を実施した．

STAMP/STPA で識別したハザード H とそれを引き起こす UCA X の組のうち，単一コンポーネントの状態として定義されるハザード H に対し，FIA を用いて X が H を引き起こすことを追試できた．ただし，一般にハザードはシステムの状態であり，ハザード H が単一コンポーネントの状態として定義出来たために，UCA X から H へ至ることが識別できたことには注意が必要である．

FTA の分析結果から説明できない STAMP/STPA のハザードシナリオが存在した．今回，FTA に必要な状態遷移では，現在の状態のみに依存して次の状態を定義した．しかし，STAMP/STPA で識別されたハザードシナリオは，同一コンポーネントが状態 A から状態 B へ遷移した後にハザードへ至るという，過去の二状態の履歴に依存して引き起こされるハザードであった．FTA 用のモデルで適切な量の履歴に依存した状態遷移モデルを構築することで，FTA によってもこのハザードシナリオを識別できる可能性もあるが，分析前に適切な量の履歴を決定することは現実的ではない．

4. おわりに

筆者らが，これからのソフトウェアプロセスに有効と考える，システム思考のモデリングの導入について報告した．ソフトウェア集約システムに有効とされている STAMP/STPA を AADL と組み合わせて活用した事例では，FTA の分析結果から説明できない結果が得られており，ソフトウェアプロセスにはソフトウェア集約型システム向けのモデリング手法の効果が高い可能性がある．このような点も含め，システム思考のモデリングはこれからのソフトウェアプロセスに有効かについて議論を行いたい．

参考文献

[1] P. H. Feiler, D. P. Gluch, and J. J. Hudak, “The Architecture Analysis & Design Language (AADL): An introduction,” DTIC Document, Tech. Rep., 2006

[2] IPA SEC, “ESPR version2 組込みソフトウェア向け開発プロセスガイド改訂版”, 2007

[3] N. G. Leveson, “Engineering a Safer World”, MIT Press, 2011

[4] S. Procter and J. Hatcliff,, “An Architecturally-Integrated, Systems-Based Hazard Analysis for Medical Applications”, Formal Methods and Models for Codesign (MEMOCODE), 2014