Pre-Formal メソッドとしての STAMP モデリング

日下部　茂†，荒木　啓二郎†

Using STAMP Modelling as a Pre-Formal Method

KUSAKABE Shigeru and ARAKI Keijiro

ねらい　形式的な仕様記述はクリティカルソフトウェアの開発に有用である一方，効果的な導入の難しさが指摘されることも多い。ライフサイクルの適切な部分に対して効果的に形式仕様記述を導入し活用するためには，具体的な仕様記述の前の段階も重要である。本発表ではそのような段階で，コンポーネント相互作用に着目するハザードのモデル化･分析手法 STAMP/STPA を用いることを提案し予備的な試行実験の結果を報告する。

キーワード　形式仕様記述，開発プロセス，ハザードモデル，STAMP

Target: Formal specifications are useful in developing critical software. In order to introduce and use formal specifications, activities and work-products before developing formal specifications are also important. We propose our approach of using a new hazard modelling STAMP as a pre-formal method. We also report our preliminary experimental results.

Keywrds:　Formal specification, Lifecycle process, hazard modelling, STAMP

１．想定する読者・聴衆

本発表は，クリティカルソフトウェアのライフサイクルプロセスのステークホルダのうち，新しい手法の導入によるテーラリングに関心があるエンジニアや管理者などを主な対象としている。テーラリングの対象はライフサイクル全体を念頭においているが，今回の発表は特にライフサイクルの上流のフェーズに焦点を当てたものとなっている。

2．背景

今後ますます，重要なインフラストラクチャでもソフトウェアで実現される割合も増え，またクリティカルな処理を含む新しいサービスがソフトウェアで実現されていくと予想される。そのようなクリティカルなソフトウェアの開発に有効なアプローチとして，形式手法を用いるアプローチがある。形式手法を用いるアプローチについて，産業界での事例の報告[1]では，形式手法の利用は成功かという問いに対し，強く同意が 61%, 同意が 34%, どちらともいえない 5%, と 9 割以上が成功と回答している。

上記のように形式手法の導入は有効とされている一方，実際の現場での利活用は限定的なものにとどまっている。筆者らは，限られた領域での先進的な形式手法の活用だけでなく，より広い領域での，形式手法の非専門家や未経験者への普及も重要と考え，ソフトウェアのライフサイクルの様々なステージにおける形式手法の効果的な導入方法について研究を行っている(図 1 参照)。形式手法も“銀の弾丸”ではなく，その導入にはポジティブおよびネガティブなインパクトがあり得，ライフサイクルの各所で，目的にかなった導入を実現するための方法論が必要と考えている。運用や保守も含めたライフサイクル全般を対象とし，システムが人や環境と相互作用を持つことも念頭において効果的な形式手法導入の検討を行っている。

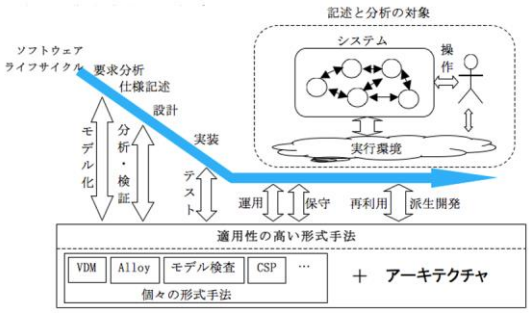

図 1　ライフサイクル各所で有効な形式手法

Fig1. Formal approaches effective at various stages in Software Lifecycle

3．課題

新たに形式手法の導入を開始する前の懸念事項のひとつに，効果的なモデル構築に対する見通しをステークホルダが持ちにくいことがある。また，形式的な仕様記述の最中や記述後にライフサイクルの複数のフェーズ，複数のステークホルダの間で形式的な仕様記述を活用しようとする際に，モデリングの意図が不明確な場合も問題になり得る。モデリングの際の捨象の基準や事前・事後条件の意図や背景が不明だと，形式的な仕様記述をもとにした実装やテストの作業が効果的

に進まない可能性がある。

このような問題を削減するには，形式手法による成果物作成開始後のモデル構築段階に焦点を当てた支援も重要である一方，形式的なモデルを構築する前の，いわゆる pre-formal な活動や成果物に焦点を当てた支援も重要である。Pre-formal な活動の支援の例としては，自然言語記述による作業成果物から形式化的な記述を開発する作業を支援する用語辞書ツールや各種 UML 図式の記述を支援する UML ツールといったものがある。

4. 提案・実験

本発表では, pre-formal なアクティビティにおいて，新しいハザードのモデリング・分析手法である STAMP(Systems-Theoretic Accident Model and Processes) [2] を活用するアプローチを提案する。

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もので，「つながるシステム」を構成するコンポーネント間の相互作用に着目した事故モデルである。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関関係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。

このような STAMP に対しハザード分析手法 STPA (System-Theoretic Process Analysis), 事故分析手法 CAST (Causal Analysis based on STAMP), STPA-Sec (STPA for Security), STECA(System-Theoretic Early Concept Analysis)といった分析法が提案されている[3]。

我々は，STAMP 事故モデルの「事故」を「受容できない損失の発生」ととらえ，形式手法を用いるライフサイクルプロセスでの pre-formal method として STAMP/STPA を利用するアプローチを提案する。STAMP/STPA はトップダウンの非形式的な手法であるものの，その提唱者の Nancy Leveson 教授は形式手法 SpecTRM の開発者でもあり，その研究室では現在も形式手法と関連する研究も実施中で，STAMP/STPA と形式手法との適合性は高いと考える。

STAMP/STPA の手順に厳密な定義はないが代表的なテキストに記載されている例を以下に示す：

- 準備 1 ： アクシデント，ハザード，安全制約の識別
- 準備 2 ： コントロールストラクチャの構築
- STPA Step1 ： 安全でないコントロールアクション（Unsafe Control Action：UCA）の識別
- STPA Step2 ： Causal factor （誘発要因）の特定

ここで，アクシデントは受容し難い損失を伴うシステムの事象，ハザードはアクシデントにつながるシステムの状態，安全制約はシステムが安全に保たれるために必要なルールであり，システムが回避すべき事象を事前に設定することで目的に沿ったハザード分析を行うためのものである。

本稿では，モデル規範型の形式手法を用いる際の pre-formal なアクティビティとして，STAMP モデリングに関して以下の二つの観点で試行と評価を行う：(1)後続する形式的な仕様記述の活動をより効率的にする，(2)複数のステークホルダ間での形式仕様記述の理解を促進する。我々は実際の現場を持たないため既存の例題を用いて以下のような試行を行った。(1)について，NPO 法人 ASTER が実施したテスト設計コンテスト 2015 の例題を用い, 形式的な記述を書く際の pre-formal method としての有効性を評価した。(2)について，形式手法ツールの一つである Crescendo のサンプル[4]に対して STAMP モデルを作成し, そのモデルの有無によって形式仕様記述の理解の容易さが異なるかを評価した。

5. 効果

(1)に関して，引き続く形式仕様記述のプロセスをガイドするか，形式的な記述を行うまでもなく行える分析ができたかで予備的な評価を行った。アクシデント，ハザード，安全制約の識別は形式仕様記述のモデリング時の抽象化や捨象に有用との主観評価を得た。コントロールストラクチャの構築はモデルのアーキテクチャに対する指針を与え，従来だと形式仕様記述によって検出していた不備の一部を事前に検出できた。

(2)に関して，STAMP モデリングの作業成果物が，初心者の形式仕様記述の理解を促進するかについて，アンケート評価を行い，その効果を確認した。

6. まとめ(今後の課題・謝辞等)

クリティカルなソフトウェアの開発での形式手法の導入をより効果的にするために，pre-formal なアクティビティとしてハザードのモデル化・分析手法 STAMP/STPA を用いることを提案した。既存の例題に対する試行と予備評価を行った結果，有用との見込みを得た。今後は実践的な事例に取り組み，客観性の高いデータを得ると同時にツール化の検討も行う。

文　献

[1] Jim Woodstock, et.al, Formal methods: Practice and experience ACM Computing. Survey. 41, 4, Article 19.

[2] Nancy Leveson, Engineering a Safer World, MIT press, 2012

[3] PSAS Home, http://psas.scripts.mit.edu/home/

[4]Claire Ingram, et.al., Crescendo Examples Compendium, Crescendo Technical Report TR-002, 2014