AdaCore
The GNAT Pro Company

TECHNICAL REPORT

# SSI：要求仕様をコードまで継承する手法のご紹介

AdaCore 技術戦略担当　M. アンソニー・アイエロ

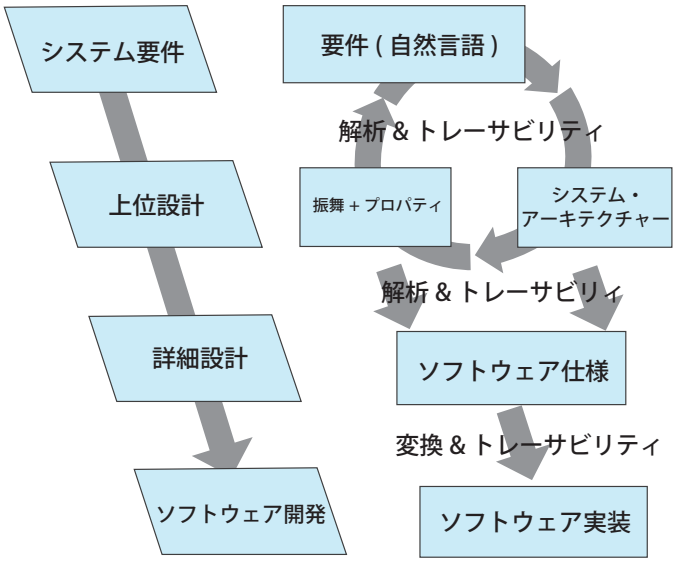
SysML で仕様を記述、Simulink へ変換、コード生成に至る開発プロセス全体にわたってシステム・ソフトウェアの整合性（Systems-to-software integrity(SSI)）を保持する事で、ソフトウェア品質を改善し、開発コストを低減、安全性およびセキュリティを向上させます。

高い信頼性が要求されるシステムやミッション・クリティカルなシステムの開発では多くの問題に直面することがあります。高機能で複雑なシステムを開発する際には、既存の開発手法では信頼性を確保できない場合があります。結果として、システム・エラーが発生し、システム障害 ( 製品寿命を短くする故障 ) を引き起こすことがあります。
多くのシステム・エラーは、重要なシステム・ソフトウェアの誤動作に起因しているため、システムに比べて、ソフトウェアの複雑さを管理するのは難しい場合があります。ソフトウェアには、物理的な範囲、重量、電力需要がありません。
システムへの機能追加は制限されることがありますが、ソフトウェアにはありません。機能が確定すれば、ソフトウェアに新たな機能を追加できます。その結果は予測でき、複雑なソフトウェアを保証することは難しく、障害を引き起こす要因となり、システム故障の原因となり得ます。

**形式化**
従来、ソフトウェアの保証は、広範なテストを通じて行われてきました。一般的なテストは、網羅的に実施できないため、残念ながら、エラーを検出して終了します。テストより、良い手法があります。それはシステム・レベルで、解析とテストを適用することです。航空機の設計を例にとってみると、設計者は、風洞での試験に入る前に、流体力学で計算して広範な空力解析を行います。基本的な保証を得るために、風洞試験では、解析モデルで検証を実施します。

ソフトウェアについては形式手法が応用できます。形式手法は、ソフトウェアの振舞を数学的に表現する技術に基づいており、ソフトウェアの状態を網羅的に検証して、エラーがないことを証明します。次に、ソフトウェア・テストは、包括的な解析であることを説明します。
形式手法の応用は特効薬とは言えず、解析を行う際に解析用の条件を用意する必要があります。そのため、重要なシステム・レベルのプロパティ（Property）をソフトウェアで明示的にたどれ（Trace）ない場合、解析条件を用意できなくなります。多くの場合、どのプロパティ（Property）を証明する必要があるのかが不明です。

SSI では、重要なシステム・レベル・プロパティ（Property）を識別し、システム・アーキテクチャを分割、成果物（Artifact）が次の成果物（Artifact）へ変換される際に、このプロパティ（Property) を保持します。システム・レベル・プロパティ（Property) が明確になると、システム設計者がソフトウェア設計者に証明すべき重要なプロパティ（Property) を引き継ぐことができるため、SSI で、形式手法を最大限に活用できます。
モデル・ベースのシステム・エンジニアリングが一般的になるにつれて、このアプローチをサポートするツールには、より強力な解析機能が対応されつつあります。



SSI ワークフロー

このツールは、待ち時間、スペース、重量、電力消費などの単純な検査ではなく、アーキテクチャの機能的な正確性を検証し、抽象度のより高いレベルで効果的に、形式手法を適用します。このツールを十分に利用するために、システム設計者は、証明すべきプロパティ（Property) を明確に識別することが必要です。SSI に焦点を当てることにより、システム設計者は、この解析ツールを十分に活用できます。
業界の設計者に確認すると、要求文書や設計文書などのシステム・エンジニアリング関連文書は、開発チームから他のチームへ「壁を越えて」引き渡されます。そのため、システム設計者からソフトウェア設計者まで、開発に関わる全ての設計者は、統合されたチームである必要があります。
人々は考えが異なるため、DO－178B/C のソフトウェア規格では、設計には高い保証が求められるため、コンポーネントを開発するチームと検証するチーム間での独立性が要求されます。ただし、チーム間に管理されたコミュニケーションがないと、信頼性が損なわれることがあります。SSI を応用すると、システムおよびソフトウェア設計者は、チーム間を仲介する文書と、重要なプロパティ（Property) を通じてコミュニケーションすることによって、システムおよびソフトウェア設計者を分けている壁を取り払うことができます。

**SSI の対応について**
SSI は、単なるツールやテクノロジーではなく、高度な保証（High Assurance）が必要なシステム・エンジニアリングの特徴を有しています。ただし、システム・エンジニアリングで SSI を応用するには、ツールのサポートが不可欠で、それは、自動変換、トレーサビリティ、解析、情報の記録の 4 つのツールに加えて、5 つ目（オプション）として、テストケース生成があります。

ツール 1 ：自動変換
成果物（Artifact）のプロパティ（Property) を次の成果物（Artifact）に変換して引き継ぐことにより、整合性 (Integrity) を保持することができます。例えば、SysML モデリング言語では、OCL (Object Constraints Language オブジェクト制約言語 ) で記述して、内部ブロック図上の出力ポートに添付された要件に関連する制約 (Constraint) を、Simulink ブロックで記述されたサブシステムの出力ポートに付属する Simulink モデルのオブザーバに変換できます。同様に、制約 (Constraint) は、ソフトウェア条件（Contract）に対応する SPARK プログラミング言語で事後条件に変換することもできます。(SysML) アーキテクチャから設計 (Simulink モデル ) または実装 (SPARK) は個別に行われており壁が存在しますが、自動変換はそのギャップを解消します。また、設計者がターゲット言語をプロパティ（Property) に従って、手動で記述する必要がないため、エラーを最小にすることができます。もちろん、変換は正確で信頼性が高いことが要求されます。この信頼性には (DO178B/C に基づくツール資格などの ) 技術が利用可能です。

ツール 2：トレーサビリティ
トレーサビリティにより、設計者は、開発時にシステム・プロパティ（Property）をたどることができるため、整合性（Integrity）を保持するのに有用です。上記の例を続けると、Simulink 同期オブザーバは OCL 制約 (Constraint) まで、たどる（Trace）ことができるため、いずれかが変更された場合でも、再検証、再生成が必要であることが通知されます。

ツール 3：解析
解析は、分割することで、設計者がプロパティ（Property）に準拠していることを証明（Prove）できるため、整合性（Integrity）を保持するのに有益です。システム開発の各フェーズで、システムが正確に記述されるまでに、高い抽象度の記述は高い精度の記述に書き換えられます。抽象度の高いコンポーネントは、複数の小さなコンポーネントに分割されるため、分割は詳細記述の精度を上げる重要な役割を果たします。
解析により、小さなコンポーネントが、抽象度の高いコンポーネントで意図された機能を保持することを証明（Prove）できます。上記の例を続けると、Simulink サブシステムは、小さなサブシステムに分割され、ソフトウェア条件（Contract）を有しているため、小さなサブシステムは上位レベルのオブザーバを満足します。

ツール 4：情報の記録
情報の記録は、トレーサビリティまたは解析が不十分な場合や、十分な対応ができない場合でも、設計者がプロパティ（Property）を継承できる情報を記録することで、整合性（Integrity）の保持が可能となります。上記の例を続けると、サブシステムの一部が商用製品（COTS）の場合、最上位サブシステムのオブザーバが満足するかを証明（Prove）できる条件（Contract）がないことがあります。ただ、商用製品（COTS）サブシステムを使用していても、他のサブシステムとともに、最上位のオブザーバを満足できることを設計者は知っている可能性があります。ツール 4 は、この情報を記録します。

ツール 5：テストケース生成
テストケースの生成は、システムが重要なプロパティ（Property）を満足していることを示すことにより、整合性 (Integrity) を確認するのに役立ちます。上記の例を続けると、サブシステムのオブザーバは、統合化試験フェーズで使用するテストケースの基を提供します。
このツールは既存の言語と既存の開発手法をサポートするため、設計者は新たな言語や開発手法を学習する必要がありません。AdaCore 社は現在 SSI をサポートするツールを開発中で、図に示すワークフローに対応します。

ハードウェア対応について
ソフトウェアの観点から SSI システム・ソフトウェアの整合性 (system-to-software integrity) に焦点を当てて解説しています。SSI はハードウェアにも同様に適用可能で、システム開発の最終段階で使用される言語は、ハードウェアによって異なりますが、全体的なアプローチは、ソフトウェア開発だけでなくハードウェア開発にも応用可能です。

まとめ
主要なプロパティ（Property）を早期に特定し、要件定義から実装までのライフサイクルを通じてシステムが達成し、保持できることが重要です。システム開発と検証に対する統合アプローチを採用することで、時間と労力を軽減します。また、システムが実行すべき事と実際に実行される事との間の不一致を防ぎます。SSI は、既存および今後のテクノロジーからの自動化されたサポートにより、社会が依存している最も重要なシステムに必要な信頼性を提供します。

Article originally published in Electronic Design Nov 13, 2018

お問い合わせ先

IT Access®
アイティアクセス株式会社

※本文中の会社名、商標、製品名等は各社の商標または登録商標です。

本社　〒222-0033 神奈川県横浜市港北区新横浜 3-17-6
TEL: 045-474-9095　FAX: 045-474-8823
E-mail: info@itaccess.co.jp URL: http://www.itaccess.co.jp

2019.9