

2011年末に発表された DO-178C 規格とその前身である DO-178B との主な相違点は、新しいソフトウェア開発手法の適用範囲を拡大した補遺部分にあります。最も重要な補遺は、モデルベース設計およびモデルベース検証の方法に関するもので、補遺 DO-331 に記載されています。これらの主要なソフトウェア設計技術は、航空宇宙産業において効率的にソフトウェアを開発し、関連する要件を保持しながら、ソフトウェアの品質と安全性を更に向上させるうえで、大きな可能性を秘めています。この記事では、DO-178C および DO-331 に則した TargetLink の使用方法および考慮すべき側面について説明します。

#### モデル：革新的手法への出発点

ソフトウェア開発を効率的かつ品質指向に基づいて行う場合、要件を DO-331 に準拠したモデルで表現することができれば決定的に優位です。単なるテキストベースの要件から、モデルとして表現された形式的要求へと移行することで、自動解析やソースコード生成、検証を行うための新しい多様な選択肢が広がります。DO-178B/C および DO-331 に準拠したソフトウェア要件には、2つの異なるタイプがあります。

#### ■ 高レベル要件 (HLR)

ソフトウェアが実行すべき動作を記述しますが、実行方法は問いません（つまり、ソフトウェアはブラックボックスとして扱われます）。HLR は、ARP4754（航空宇宙推奨プラクティス）などに基づくシステムプロセスで定義された実際のシステムに関する要件から導出されます。

>>

```

        description: number of axis#1 points */
        6 /* Ny:
description: number of axis#2 points */
        (const uint16 *) &(Ramp_Rate__Ki__x_tabl
        (const uint16 *) &(Ramp_Rate__Ki__y_tabl
        (const uint16 *) &(Ramp_Rate__Ki__z_tabl
    };

    * SLStaticLocalInit: Default storage class
    *
    static sint32 X_Sc4_Discrete_Time_Integrator
    , 1.99993896484375 */;

    * BusInport: TL_FuelsysController/Run_Airf
    te_Read_RpCorrectedSensors_Sensors(&Sensor
    * # combined # TargetLink outport: TL_Fuel
    * # combined # Discrete Integrator: TL_Fue
    egrator */
    te_IrvIWrite_Run_AirflowCorrection_Airflow
    ((sint32) 300));

    * Discrete Integrator: integration
    * # combined # Product: TL_FuelsysControll
    * # combined # 2D-TableLookup: TL_FuelsysC
    * # combined # Sum: TL_FuelsysController/R
    * # combined # Relational: TL_FuelsysContr
    _Sc4_Discrete_Time_Integrator += ((sint32)
    Tab2DS17I2T4169(&Sc4_Ramp_Rate__Ki__map, S
    ((sint16) (((uint16) (Sensors.Ego <= 3199

```



航空宇宙産業向けアプリケーションに  
最適なコード生成ツール TargetLink

# Safe Code

According to

# DO-178C

dSPACE の量産コード生成ツール TargetLink は、自動車向けの量産プロジェクトだけでなく、民間機や軍用機のプロジェクトにも適しています。dSPACE では、特に DO-178 に準拠した航空宇宙開発プロジェクトでの TargetLink の使用に関する総合的なワークフローの記述を提供しています。このワークフローでは、TargetLink ベースのツールチェーンを使用してソフトウェアの認証を容易に行う方法について説明しています。



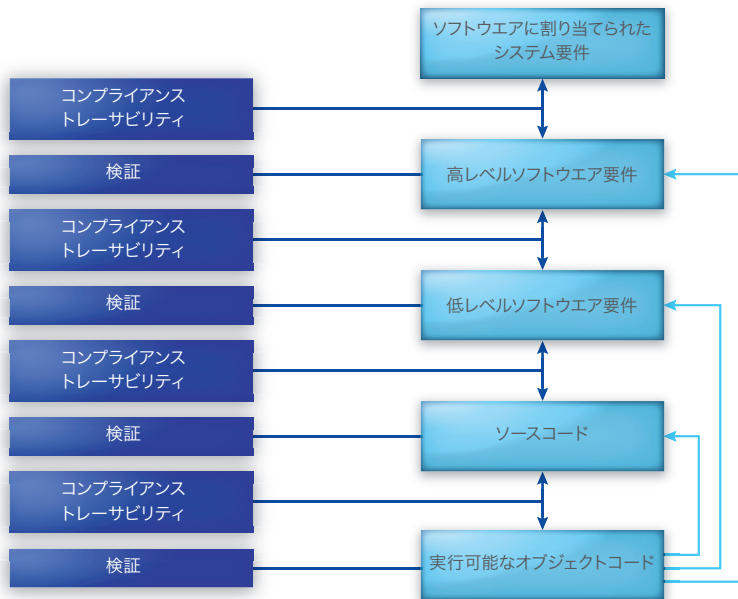


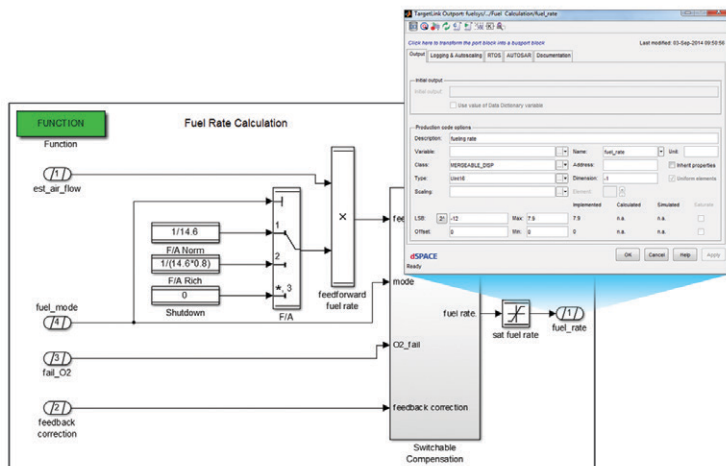
図 1：必要な検証手順を含む、DO-178C に基づいた重要な開発フェーズ。

#### ■ 低レベル要件 (LLR)

ソフトウェアの内部動作を記述します (ホワイトボックスとみなされます)。つまり、実行すべき動作と実行方法を記述します。LLR は、当然ながら HLR から導出されます。実際のソースコードは、LLR から直接生成できる必要があります。

モデルを使用すると、これらの 2 つのレベルにおける要件を表現することができます (図 1)。LLR を表現するためには、多くの場合、Simulink®/TargetLink® モデルが使用されます。実際のソースコードは、自動コード生成によって LLR から生成されます。DO-331 では、LLR を表現するこの

図 2：Simulink/TargetLink 設計モデルを使用すると、TargetLink でソースコードを直接かつ自動的に生成することができます。



ようなモデルを設計モデルと呼びます。このモデルには、実際の機能の記述に加えて、内部データ構造や制御フロー情報、可能な固定小数点表現など、ソフトウェアに関する必要なすべての詳細情報が含まれています (図 2)。

#### クリック 1 つで設計モデルからソースコードを生成

DO-331 に基づいて要件を表現する設計モデルでは、ソフトウェアのソースコードの作成は、手作業によるプログラミングではなく、コードの自動生成によって行うことができます。TargetLink を使用すると、ボタンをクリックするだけで確実にソースコードを生成することができ、その品質と信頼性は手作業でのプログラミングを遙かに凌駕しています。

- TargetLink で生成されたソースコードは、非常に可読性が高く、レビューに適しています。これは、ソースコードへの詳細なコメントや理解しやすいシンボル名、および C 言語のサブセットが使用されているためです。
- コードは、設計モデルにそのままトランスラシすることができ、ソースコードおよび生成に使用された関連モデル間のトレーサビリティを直接保証することができます。
- さらに、TargetLink で生成したコードにさまざまな設定を行うことで、コーディングガイドラインの遵守を確認したり、TargetLink で生成したコードと既存のレガシーコードを組み合わせたりすることも可能です。生成されたコードをソフトウェアアーキテクチャに最適な形で統合することもできます。

TargetLink 生成コードが提供する優れた品質、設定の柔軟性、および効率性は、すべての適用分野で幅広く認められています。

#### モデルベース検証：

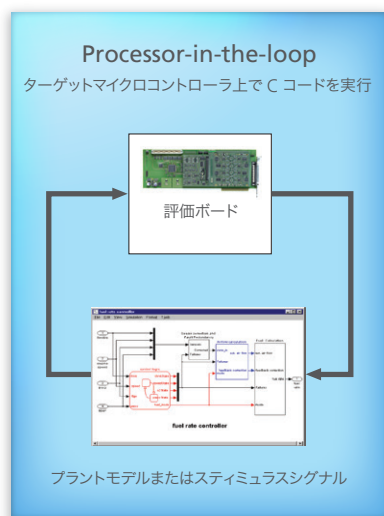
##### 認証を容易に行うための重要な鍵

(HLR や LLR といった) 要件の指定にモデルを使用することの優位性は、量産コードの自動生成だけでなく、検証手順などの他の領域でも明らかです。モデルやソースコード、オブジェクトコードといった生成物を個々の開発段階でテストするためには、これらの手順を開発プロセスと並行し

て実行する必要があります(図1)。モデルが必要な要件を満たしているか検証するには、モデルシミュレーション、カバレッジ分析、およびテストケースの生成を組み合わせて使用します(図1)。テストケースは、DO-178B/Cに基づいて要件ベースのみで作成する必要があります。要件自体がSimulink/TargetLinkなどのモデルとして表現されている場合、BTC EmbeddedTester<sup>®</sup>などの自動テストベクトル生成技術を使用することができます。実行可能なオブジェクトコードがHLRやLLRと一致しているか検証する標準的な方法は、そのコードをターゲットプラットフォームで実行することです(図1)。TargetLinkは、このための非常に強力なメカニズムをPIL (Processor-in-the-Loop) シミュレーションの形式で提供しています。TargetLink上では、自動生成されたコードがターゲットコンパイラによって直接変換され、ターゲットプロセッサを搭載した評価ボード上で実行されます(図3)。

### TargetLinkに関する DO-178C/ DO-331 ワークロードドキュメント

dSPACEでは、DO-178C/DO-331に準拠したプロジェクトでのTargetLinkの使用に関するワークロードドキュメント『TargetLink – Model-Based Development and Verification of Airborne Software』を提供しています。



DO-178C/DO-331の個々の要件、すなわち「目的」を満たす方法については、ワークロードドキュメントに記載されています。このドキュメントでは、TargetLink自体に注目するだけでなく、サードパーティ製ツールを含む完全なモデルベースのツールチェーンで構成されるTargetLinkエコシステム全体にも焦点を当てています。これには、BTC Embedded Systems社、Model Engineering Solutions社、およびAbsInt社といったTargetLinkパートナーから提供されるツールも含まれています。このドキュメントを入手するには、TargetLink.Info@dSPACE.de宛てに電子メールでお問い合わせください。■



## まとめ

TargetLinkは、DO-178Cに準拠した航空宇宙開発プロジェクトに最適であり、ボタンをクリックするだけで高品質なソースコードを生成することができます。レイアウトへのコメント機能やシンボル名により、TargetLinkで生成されたコードは再利用性が高く、要件をシームレスにトレースしたり、コーディングガイドラインへの遵守を確認したりすることも容易です。TargetLinkにサードパーティ製ツールを統合すれば、検証、シミュレーション、解析、およびテストを行うための理想的な環境が実現します。TargetLinkにより、開発プロジェクトの要件から最終的なソースコードまで、DO-178Cに準拠した形で管理することができます。

図3：PIL (Processor-in-the-Loop) シミュレーションで実行可能なオブジェクトコードを実行し、このオブジェクトコードが要件に適合するかを検証します。