



Shift Gears Quickly

モデルベース開発における TargetLink を使用したトランスミッション
制御ソフトウェア開発 (Continental 社)



プロジェクトの目標は、プロトタイプと最終製品の開発時間を大幅に短縮することです。この目標を達成するには、組み込みソフトウェアを開発するための革新的な手法が必要です。そのため数年前に、Continental社は、トランスミッション電子制御ユニット (ECU) の開発にモデルベース開発および量産コード生成を導入することを決定しました。この記事では、プロセスと手法の作成から、2つの量産プロジェクトにおけるコード生成ツール TargetLink の効果的な使用まで、その全体像を示します。



デュアルクラッチトランスミッションは、マニュアルトランスミッションのエネルギー効率とトルクコンバータトランスミッションの快適性を兼ね備えています。

この 10 年間で、組み込みソフトウェア開発の需要は大幅に増大しました。従来の手作業によるプログラミングおよびソフトウェアテストに加えて、現在は、安全基準と開発手法に関するより厳しい要求が課題となっています。ソフトウェア開発が複雑になるにつれて、これらの要求は、デュアルクラッチトランスミッション (DCT) と無段変速トランスミッション (CVT) を含む新世代トランスミッションでは一段と厳しくなっています。さらに、自動車業界のサプライヤは、市場の要求に応えるために、開発時間を短縮する新しい方法を常に模索しています。こうした状況を受け、Continental 社のトランスミッション事業部では数年前に調査を実施し、アプリケーションのモデルベース設計および量産コード生成について分析しました。この革新的なモデルベース手法による生産性の向上について、まず、その手法を社内プロジェクトで使用してみることで評価しました。同時に、コードの自動生成を含むモデルベース設計のプロセス、手法、ツールを決定し、トランスミッション事業部の固有のニーズに合わせて調整しました。最初の量

産プロジェクトでの使用が成功した後は、このツールチェーンを使用してデュアルクラッチトランスミッションのアプリケーションソフトウェア全体を開発しました。この記事では、個別の開発ステップを説明し、さらに初期テスト、最初の量産プロジェクト、コード全体を TargetLink で自動生成したデュアルクラッチトランスミッションから得た経験についても説明します。

プロセスと手法の設計

社内プロジェクトの一環として、モデルベース設計および量産コードの自動生成を開始するために、最初に必要なプロセスの手順を定義しました。制御ロジックとソフトウェア開発の V サイクル全体に対応するモデルベース設計では、自動車メーカー、トランスミッションサプライヤ、電子機器サプライヤ間のやり取りに関して、次のような要件をまとめました (図 1 参照)。

- ECU の物理的要件のモデル化
- 実際の ECU 動作のシミュレーション (タスク、実行順序、オペレーティングシステム)

- マルチユーザ環境の確立による分散開発のサポート
- モデル、コード、スクリプトを従来のプロセスに適合させることによる検証、妥当性確認、およびアーカイブ

最初に、モデルベース開発手法をデモ車両の機能に適用しました。そこで採用されたソリューションは、最新の技術に基づいて作成されたツールを使用することでした。そのため、機能設計には MATLAB®/Simulink® を選択しました。プロジェクトの成功は、デモ車両における新機能のデモンストレーションが非常に短期間で完了できるかどうかにかかっていた。コンセプトのシミュレーションと、それに続くテストおよび検証は、PC 上で MIL (Model-in-the-Loop) シミュレーションを使用して実施しました。検証は成功し、ラビッドコントロールプロトタイピング (RCP) 環境を備えたデモ車両とテストベンチで機能をテストする準備ができました。機能の妥当性確認と微調整には、Real-Time Workshop® と dSPACE AutoBox/MicroAutoBox を使用しま

「TargetLink を使用することで、製品化までの時間が一層短縮され、お客様にご満足頂くことができました」

Georg Grassl, Continental AG

した。初期テストによって、お客様が提示した要件をごく早い時期に評価できるようになりました。ラピッドプロトタイピングの段階におけるモデルベース開発が成功した後は、Vサイクルの段階を、手作業によるCプログラミングから量産コードの自動生成に切り替えました。このステップでは、コード生成ツールとしてdSPACE TargetLinkを選択しました。

16ビットマイクロコントローラ用に最適化されたコードの自動生成が実現可能であることを証明するために、トランスファーストランスミッションの位置制御アルゴリズムを使用しました。最初に、既存の制御ロジックをモデルベース設計に移行し、量産コード生成の実装情報を使用してそれを拡張しました。次に、自動生成されたコードを統合して、量産ECU上で妥当性を確認しました。生成されたコードの効率性を証明するために、そのリソース消費に

ついて、量産車で既に使用されている手作業でコーディングされたソフトウェアと比較しました。

最後に、これらの予備テストに基づいて、新しい開発手法で量産ソフトウェアを生成するために重要な工程要素を確認しました。次のセクションでは、いくつかの支援手法を含むこれらの工程要素について、より詳しく説明します。

モデルベースの機能設計

浮動小数点モデルのシミュレーションによって、機能要件に適合できるかどうかを迅速に評価できるようになりました。このステップの後には、機能要件の妥当性をデモ車両で確認するためラピッドプロトタイピングが続きます。最初に、機能アーキテクチャと、それに関連して生成されるモデルライブラリを定義して、分割開発および構成要素の作成が可能な状態しておく必要があります。さらに、制御ロジックと

ソフトウェアのモデリングガイドラインも、互換性のあるモデルベース設計を実現するのに役立ちます。工業規格は社内ガイドラインとの整合性が評価され、必要であればプロジェクト固有またはアプリケーション固有の規則が追加されます。

モデルベースのソフトウェア設計

最適化されたターゲットコードを生成するモデルベース設計のテンプレートを導入して、機能設計の段階からテンプレートに準拠する必要があります。このことは、MISRA (Motor Industry Software Reliability Association) とツールメーカーによって規定された規格 (MISRA AC TL ガイドラインなど) と密接に関連しています。ソフトウェアの設計では、スケーリング、ブロックプロパティの定義、コード生成ツールの設定調整を行って、コードの自動生成をサポートします。いわゆる実装モデルは、実装情報 (メモリ割り当てなど) が追加されるときに作成されます。

この時点で、ソフトウェアアーキテクチャと機能アーキテクチャを一緒に早期に定義する必要があります。この結果としてCモジュールとC関数への分割が行われ、テスト、構成 (アーカイブ)、および保守が可能になります。

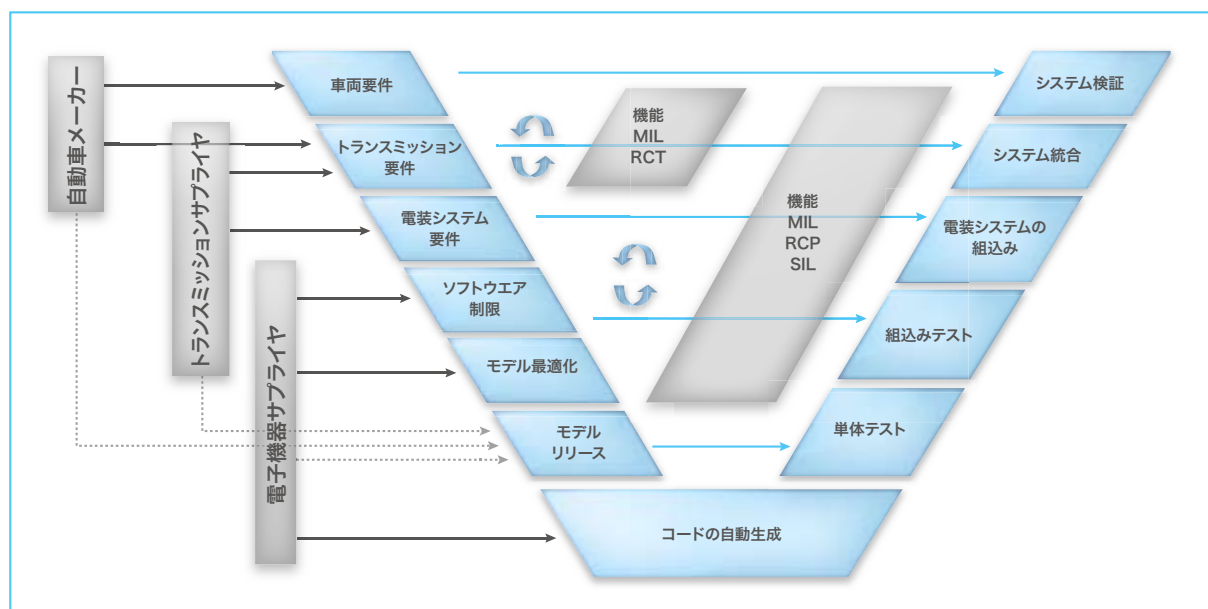


図1: トランスミッション事業部での開発サイクル

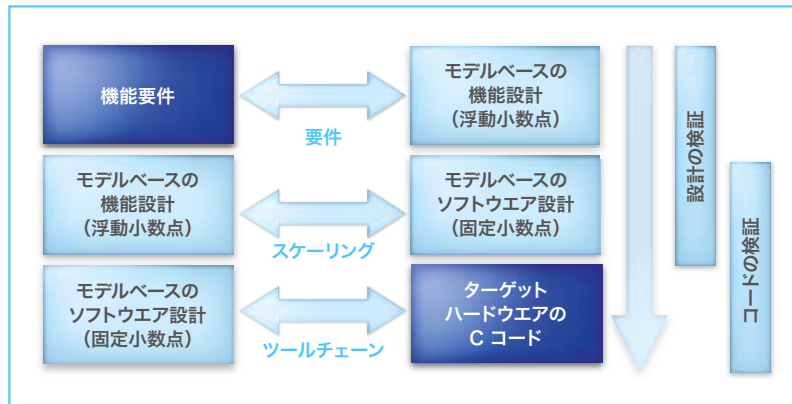


図2: モジュールベースのテスト - 設計とコードの検証

モデルベースの連携

一貫性のある方法でモデル設計と交換を行うために、開発チーム全体がモデルレベルで密接に連携する必要があります。複雑な機能はより小さい単位に分割されて、モデルライブラリに格納されます。従来からある手書きのソフトウェア要件の仕様書は、モデルに基づいたドキュメントに置き換えられます。

モデルベースのテスト

特に自動化によってプロセスを最適化する

新しいプロセスと手法による最初の量産アプリケーション

上記の予備調査の結果に基づいて、モデルベース設計とコードの自動生成に関するすべての分野のパートナーと緊密に連携した最初の大規模プロジェクトが開始されました。モデル化されたアルゴリズム（アプリケーションソフトウェアのサブ機能）は、6速オートマチックトランスミッション用のギアシフト方式（図3参照）で、16ビットプラットフォームで使用されます。予備調査が成功したことで、事前に開発されたプロ

コードに関する社内規則との互換性を保証するために、静的なコード分析を行いました。また、モデルレベルでは、構造カバレッジテストがテストベクトル生成のツールによって実施およびサポートされました。この結果、固定小数点コードの100%カバレッジ（テストの最高カバレッジレベル：変更された条件/決定カバレッジ、MC/DC）を実現するという品質目標を達成しました。これらのモジュールテストは、マイクロコントローラプラットフォームをシミュレートできる標準ツールを使用して実施しました。これによって、完全なツールチェーン（コード生成ツール、コンパイラ、リンカ）を検証することに成功しました。

最初の量産プロジェクトの結果とその考察

私たちが選択した手法では、厳しい時間枠を設けることで、作業を要求された時間内で実施することができました。この結果、このプロジェクトにおける製品化までの時間が一層短縮され、お客様にご満足頂くことができました。さらに、開発したトランスミッションソフトウェアでは、次のようなリソース消費に関する厳しい制限も守る必要がありました。

- CPU 使用率は最大 15%
- ROM 割り当ては最大 100KB

TargetLink を使用した場合、値はこれらのレベルよりもはるかに低くなり、自動生成された制御ソフトウェアでは次のようになりました。

- 最大 10% の CPU 割り当て
- 60KB のコンパイル済みコード
- 50 個のモジュール
- アプリケーションソフトウェア全体の 20% の ROM 割り当て

「自動車メーカーが社内調査として検証を行ったところ、手書きのコードのエラーの少なさは TargetLink によって生成されたコードに及ばないという結果になりました」

Georg Grassl, Continental AG

場合は、モデルベースのテストとそれに対応する品質測定が不可欠です。そのため、一般的なソフトウェア品質測定を修正して使用します。これによって短時間の極小 V サイクルが可能になり、モデルを短期間で完成させることができます。モデルベースのテストは、段階的なアプローチを可能にします（図2参照）。

セスと手法に従って MATLAB/ Simulink および dSPACE TargetLink を使用しました。

制御ソフトウェアのテストは、標準的な品質保証手順に従って実施しました。この手順は、記述済みの C コードに対して既に定義されており、またコードの自動生成の条件に適合するように調整されていました。

遅くともソフトウェア品質測定とコード検証の開始までに、適用するコード生成ツールのバージョンを確定する必要があることが明らかとなりました。コード生成ツールに関するすべてのアップデートはソースコードの変更につながる可能性があり、変更が発生した場合はすべてのソフトウェアテストと品質測定を再実行する必要があります。自動車メーカーが社内調査として検証を

行ったところ、手書きコードのエラーの少なさは TargetLink によって生成されたコードに及ばないという結果になりました。モデルベース開発とコードの自動生成を使用したことで、私たちは非常に多くのソフトウェアをスケジュール通りに納品することができるようになりました。これは、この新しい手法の構造化されたプロセスのおかげです。この手法は別の場所にいるユーザに送られ、トランスミッションの開発プロジェクトで標準の手法として使用されました。

デュアルクラッチトランスミッションの制御ソフトウェアの開発

2006 年に、新しい手法とプロセスを使用したときの貴重な経験のおかげで、デュアルクラッチトランスミッション ECU のハードウェアとソフトウェアの生産開発を開始することができました (図 4 参照)。この制御ソフトウェアの全ソースは、モデルベース開発と TargetLink によるコードの自動生成を使用して作成しました。作業の分担は次のようにしました。

- 自動車メーカーは機能要件を作成します。
- Continental 社は、ラビッドコントロールプロトタイピングを使用して妥当性が確認されたモデルベースの機能設計を利用できるようにします。



デュアルクラッチトランスミッションの電子制御ユニットは、Continental Automotive システムラボの認定試験に合格する必要があります。

- Continental 社はトランスミッション ECU のサプライヤとして、ハードウェアとソフトウェアを利用できるようにします (モデルベースのソフトウェア設計、コードの自動生成、モデルベーステスト、品質保証を含む)。
- 自動車メーカーと Continental 社は、HIL (Hardware-in-the-loop) シミュレーションによって、車両におけるシステム統合および機能の妥当性確認を実施します。
- 制御ロジックの再利用
- デバイス固有の適手法へのコード生成ツールの適応
- マルチユーザ環境での分割設計

ドの自動生成の取り扱いは、最初の量産プロジェクトでの取り扱いと同様でした。プロジェクトの実行中に、DCT の複雑な機能が原因で、さらに次のような課題が発生しました。

次の 3 つのセクションでは、これらの課題にどのように対処したかを簡単に説明します。

この DCT プロジェクトでの工程要素、モデルベース開発の対応する手法、およびコー

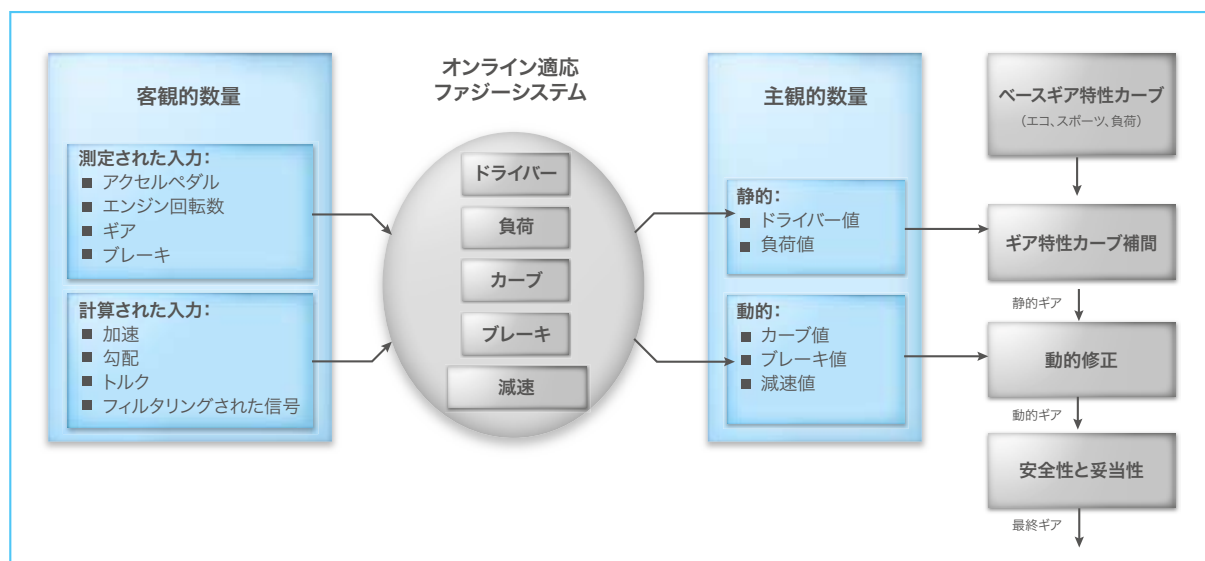


図 3 : ギアシフト方式 - 構造と信号フロー

制御ロジックの再利用

DCTシステムを調べてみると、2つのクラッチと4つのギア装置で制御ロジックの再利用が必要なことは明らかです（図4参照）。さらに、限られたハードウェアリソースが再利用の必要性をより大きくしています。2つのクラッチ、2つのシャフト、および4つのギアアクチュエータのアクチュエータ制御ロジックには、共通のアルゴリズムを使用する必要があります。

また、このプロジェクトでは下位の制御ロジック（つまり、フィルタルーチン）にも Simulink ライブラリを使用しているため、「ネストされた再利用」が行われます。リソース消費の削減を可能にした TargetLink の制御ロジック再利用機能によって、この要件は満たされます。開発者は制御ロジックが再利用できることを知っているため、モデルアーキテクチャを設計する際にそのことを考慮に入れることができます。

デバイス固有の適合手法へのコード生成ツールの適応

このプロジェクトでは開発コストを削減するために、電子制御ユニットに固有の適合手法を、適合デバイスやメモリ拡張を使用せずに実装しました。この結果、一連の適合ツール用データは構造体として編成され、実行時におけるこのデータへのアクセスは、構造体へのポインタをROMからRAMへ再ルーティングすることで行われるようになりました。必要なコード

「モデルベース開発とコードの自動生成を使用したことで、非常に多くのソフトウェアをスケジュール通りに納品できるようになりました」

Georg Grassl, Continental AG

パターンを実装するために、一方では、バリエーションコーディングやテンプレートなどの TargetLink 機能を使用しました（図5参照）。その一方で、私たちは dSPACE との共同作業で、コード生成ツールに変更を加えました。DCT モデル全体をこれらの固有の適合手法に変換する作業は、1週間以内で完了しました。これは、コード生成ツールによって提供されるオープンプログラムインターフェースが本当の意味で高性能であることを証明するものです。

マルチユーザ環境での分割設計

このプロジェクトにおける機能設計は、Continental 社だけではなく、関係する自動車メーカーでも実施されました。以前に使用していたシングルユーザ手法では関係する多数の開発者に対処することが困難であったため、開発したプロセスをマルチユーザに対応できるように拡張する必要があります。具体的には、大規模な機能をより小さなモデルフラグメントに分割して、Simulink モデルライブラリで管理

しました。環境内でデータの一貫性を保つために、dSPACE データディクショナリのインクルードファイルによって提供されるマルチユーザサポートを使用しました。トランスミッションチームは、モデル、データ記述、テストベクトル（機能および構造に関する）、テスト分野、およびファインチューニングのガイドラインを、設定管理のパッケージとして保持することを決定しました。これによって、他のプロジェクトでの再利用、および一貫性のある開発を円滑に行うことができるようになりました。

結果および重要な結論

事前に開発したプロセスと手法のおかげで、DCT プロジェクトの制御ソフトウェアの 100% をモデルに基づいて開発し、TargetLink でコードを自動生成することができました。この大規模プロジェクトで作成された制御ソフトウェアを量的に示す数字は、次のようになりました。

- 250KB のコンパイル済みコード
- 120 個のモデルライブラリ

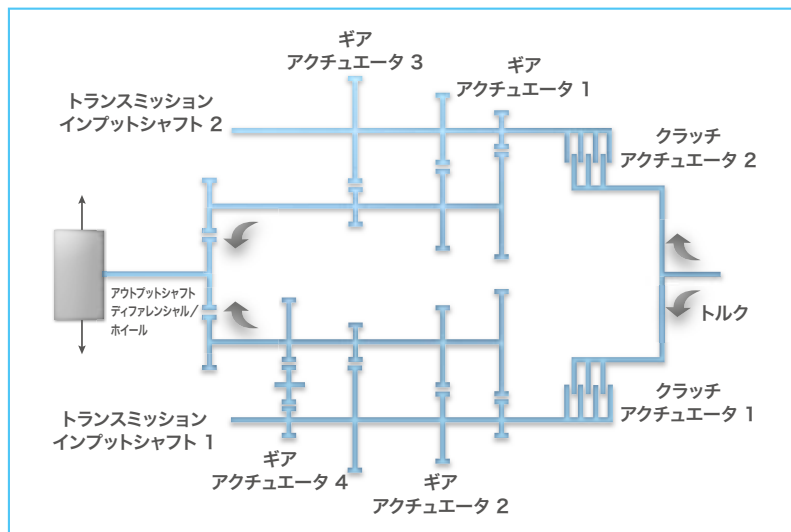


図4：DCTの概略図

上記の課題とそれに対応するソリューションについては、コード生成ツールの提供元と信頼できる協力関係を築けたことが大きかったと思います。これは特に、メモリ割り当て、データバリエーション、およびコード生成ツールがどれだけ堅牢であるかについて、プロジェクトチームが特定のサポートを依頼した場合に言えることでした。またコードの自動生成を含むモデルベースの開発で、コード生成ツールのアップデートとバッチをプロジェクトのソフトウェアのタスクと同期させる上でも非常に重要なことでした。



図5：変数テンプレートを使用してデータバリエーションへのポインタを定義し、バリエーションをパラメータに割り当てます。

今後の展望

モデルベース設計およびコードの自動生成にプロジェクト固有のアプローチを使用して得たノウハウによって、現在トランスミッションチームではこの手法を下位ソフトウェア（アクチュエータ制御）にいつでも使用できるようになっています。さらに、迅速な手法を引き続き整備する一方で、プロジェクト固有のアプローチからプラットフォーム単位のアプローチへ移行する作業も進めています。これによって、私たちは一貫性を維持しながら、この開発手法の導入を全世界規模でサポートすることができるようになります。■

Georg Grassl
Business Unit Transmission
Gerd Winkler
Business Unit Engine Systems
Continental AG
ドイツ

まとめ

トランスミッション事業部では、モデルベース開発とコードの自動生成のために、シームレスな手法を実装しました。開発フェーズの各段階は、実行可能な仕様（モデル）によって密接に関連づけられています。また一方で、モデルベース設計とコードの自動生成を使用して、DCTの制御ソフトウェア全体を実装しました。自動車メーカーとの共同作業によって、トランスミッション事業部は制御ソフトウェア開発の効率的なプロセスを確立できました。

この手法に使用したツールチェーンは、既に大規模プロジェクトでその価値を実証済みでした。プロセスは、通常の開発手法を使用するプロジェクトからの信頼できる要素に基づいており、モデルベースの開発およびコードの自動生成に関する要件に適合するように調整されました。これによって、新しい手法を効率的に導入できるようになりました。

プロセスは段階的に実行されます。最初に、社内プロジェクトで新しい手法を実装します。次に、初期量産プロジェクトで、アプリケーションソフトウェアの20%のコードを自動生成します。最後に、非常に複雑なデュアルクラッチトランスミッションのアプリケーションコードの100%を自動生成します。

この新しい手法を使用したことで、ただちに品質が向上しました。同様の複雑なプロジェクトと比較した場合、従来の開発手法を使用するプロジェクトと、モデルベース開発手法（ラビッドコントロールプロトタイピング、量産コードの自動生成、モデルベーステストなどのオプションを含む）を使用するプロジェクトでは、モデルベース開発の方が従来の手法よりも明らかに優れています。

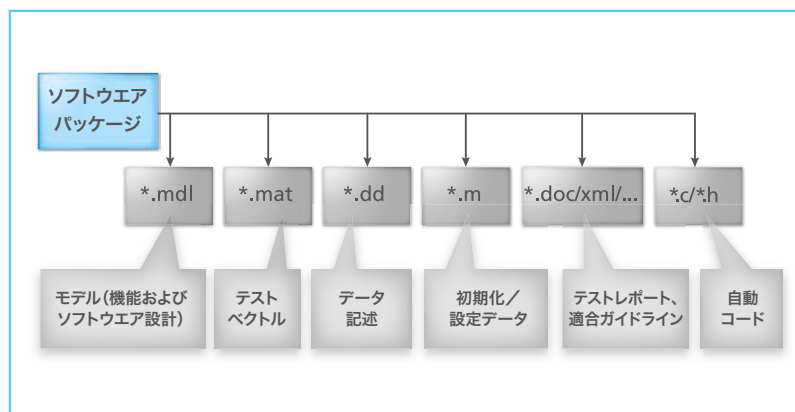


図6：マルチユーザ環境のソフトウェアパッケージ