

ソフトウェア・プロダクトライン開発における テストプロセス自動化フレームワーク

キーワード

プロダクトライン開発、フォーマルメソッド (形式手法)、
テストベクタ自動生成、要件からテストへのトレーサビリティ

はじめに

近年企業は、大規模・複雑さを増すソフトウェア開発のスピードを加速し、生産性向上によるコスト削減、高い品質を確保するために、あらゆる開発成果物を体系的に管理して再利用するといった工業化策である、ソフトウェア・プロダクトライン (SPL) への取り組みが戦略として求められている。

また高信頼性システムにおいて、ソフトウェア開発プロセスの70%をも占めるテストの多くは、手作業に頼っている。

さらにこれらテストは、デザインレビュー、コードの静的解析、ユニット/インテグレーション/システムテストなど、開発ライフサイクルの各フェーズで分断されており、要件やコードなどの修正・変更があった場合の追跡・対応が極めて困難となっている。

プロダクトライン開発において、テストの自動化と再利用を促進すること、また要件からのトレーサビリティにより、全体を可視化して管理を効率化させ、迅速かつ正確に検証できる仕組みが必要である。

これらの課題に対し、開発組織は独自手法や各種ツールによる対処を試みているが、ツールの断片化により開発フェーズ間の垣根が埋まらないのが現状であり、体系的な再利用や自動化が進んでいない。そこでこれらを支援するツールの統合を図り、ソフトウェア・プロダクトライン・ライフサイクルに於けるテストプロセス自動化を調査した。

1.1. プロダクトライン開発管理支援ツール

pure::variants は、従来のフィーチャ・モデルベースの手法ではなく、製品の機能・課題をフィーチャ・モデルに、プラットフォームなどソリューションをファミリー・モデルに管理する。複数のモデルと階層からなる大規模システムをサポートし、製品ファミリーの全ての変動要素、共通要素が体系的に再利用できるようになる。また、オープンインターフェイスを提供し、要件管

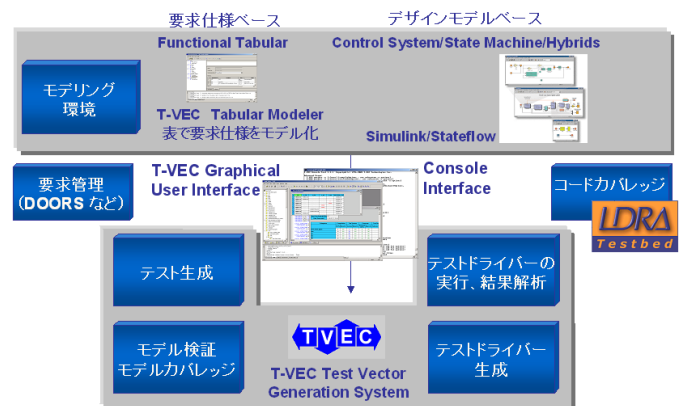
理、モデル駆動開発、構成管理、テスト・バグ管理など、あらゆるツールをプロダクトライン対応にすることが可能となる。そして、フィーチャの選択からバリエーションモデルを設定し、変換することで、バリエーションごとの要求仕様書、デザイン、ソースコードなどが自動生成される。 [1]

1.2. モデルベース検証・テストベクタの自動生成

T-VEC は検証に先駆けて要求モデル (TMM) やデザインモデル (Simulink/Stateflow) を独自の階層的 Disjunctive Normal Form (DNF) に変換する。この形式に於ける各機能要求 (入出力間の振舞い) の入力域に対するコンストレインツは Domain Conversion Path (DCP) として抽出される。このパスに対し、テストベクタ生成システムは、上流の制約を受けて伝播されてくる入力範囲で期待出力との組合せを生成する。この組合せ (テストベクタ) を生成できない DCP は、モデル上の矛盾として検出される (モデル検査)。また、このテストベクタ生成機能は、フロート、ダブルなど各種データ型、リニア/ノンリニア式に対応しているため、矛盾無く生成されたテストベクタは、実システムに対する入力と期待値としてテストに用いられ、モデルに対するコードの一致性の検証が行える。T-VEC は、以下の機能で構成される。

- T-VEC Tabular Modeler (TMM)
- Simulink Tester (SL2TVEC)
- T-VEC Vector Generation System (VGS)

TMM は、表形式で要件のモデリングをサポートする。SL2TVEC は、デザインモデルである Simulink や Stateflow を変換し解析する。VGS は、これらモデルの解析、テストベクタ生成、テストドライバー生成、テスト結果解析、結果レポート生成などを行う。

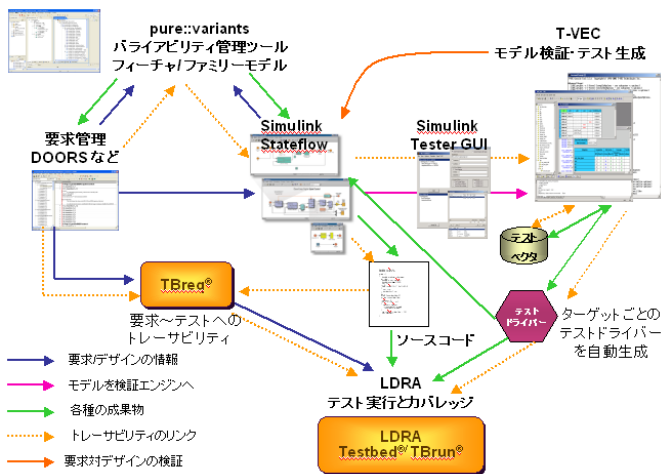


T-VECは、フォーマルメソッドを用いたモデルベース検証ツールで、高信頼性ソフトウェアのV&Vをサポートする目的で開発された。航空機開発におけるガイドラインFAA/DO-178B,Cに関わり、モデルベース開発とその検証についてのガイドライン制定に貢献するなど実践的に産業界で使用されるツールである。最近では、NASAの次世代有人宇宙船開発(Orion)に採用されている。

1.3. テストプロセス管理支援フレームワーク

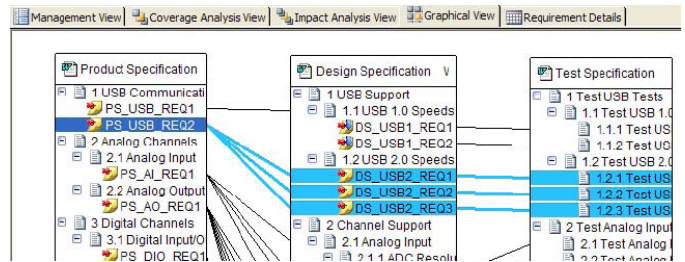
LDRA ツールには、テストプロセス管理支援フレームワークであるTBreqに、静的解析/カバレッジ解析を行うTestbed、そして単体テスト、リグレッションテストを自動化するTBrunが有機的に統合されている。これらテスト機能は、ホスト・シミュレーションから実ターゲット実行まであらゆる環境をサポートする。そのため、要求から実システム動作環境のテストに至る、広範なテストプロセス全ての自動化を支援し、開発者・テスト担当者は、開発初期段階で欠陥を識別して要件への照会ができる、高度に洗練された統合環境を享受できるようになる。そのため、要求仕様ベースのテストが実ターゲット・システムで行われたエビデンスが求められる、航空・宇宙・防衛システムなど高信頼性マーケットで30年以上に渡って採用されている。

2. SPL とテストプロセス自動化フレームワークの統合



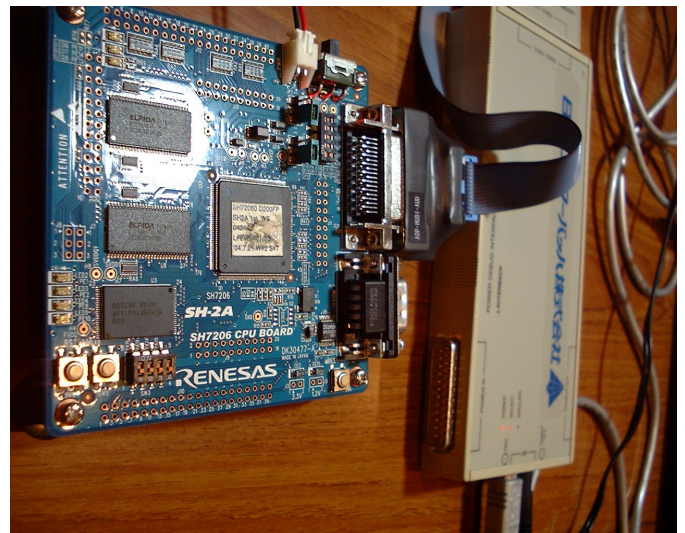
上述のツールを統合し、プロダクトライン開発で体系的に管理される資産からバリエーションを自動生成し、テストプロセス自動化フレームワークを用いて要件がテストされた証明を取得するまでの統合化を確認した。おおよその手順は、要求仕様書(DOORS)が、pure::variantsのフィーチャ・モデルに、デザインモデル(Simulink)はファミリー・モデルに登録され、互いの相関を設定する。そしてフィーチャの選択により、バリエーションごとの要求仕様書、デザインモデルが自動生成される。生成されたモデル(Simulink)に対し、T-VECのテストベクタ生成を試みることで、

モデル上の欠陥・矛盾を解析する。そして、LDRAテストプロセス管理支援フレームワークを用いて要件ごとにソースコードを割り振り、テストを実行することで、静的解析・テストベクタの実行・カバレッジ結果が、自動的に要件にマップされ、トレーサビリティのエビデンスを得る。



要件からテストへのトレーサビリティマトリクス

テストの実行は、ホスト・シミュレーション、実ターゲットなど、実行環境に合わせたテストハーネスを入れ替えるだけで、バリエーションの成果物、プロセス等をそのまま再利用できることを確認した。(ルネサスSHターゲット、ローターバツン社デバッグ)



従来型のテストツールは、開発ライフサイクル内の限られたフェーズにのみフォーカスされたものが多いが、SPLでは、複数のバリエーションが同時に開発され、その中でプラットフォームなどコア資産の変更が頻繁に行われる。調査したテストプロセス自動化フレームワークならば、あらゆる資産の変更の影響を、迅速かつ正確に検証することが可能となる。

[1] Software Product Line Engineering with Feature Models
 Danilo Beuche, Mark Dalgarno,
<http://www.pure-systems.com/fileadmin/downloads/pure-variants/tutorials/SPLWithFeatureModelling.pdf>