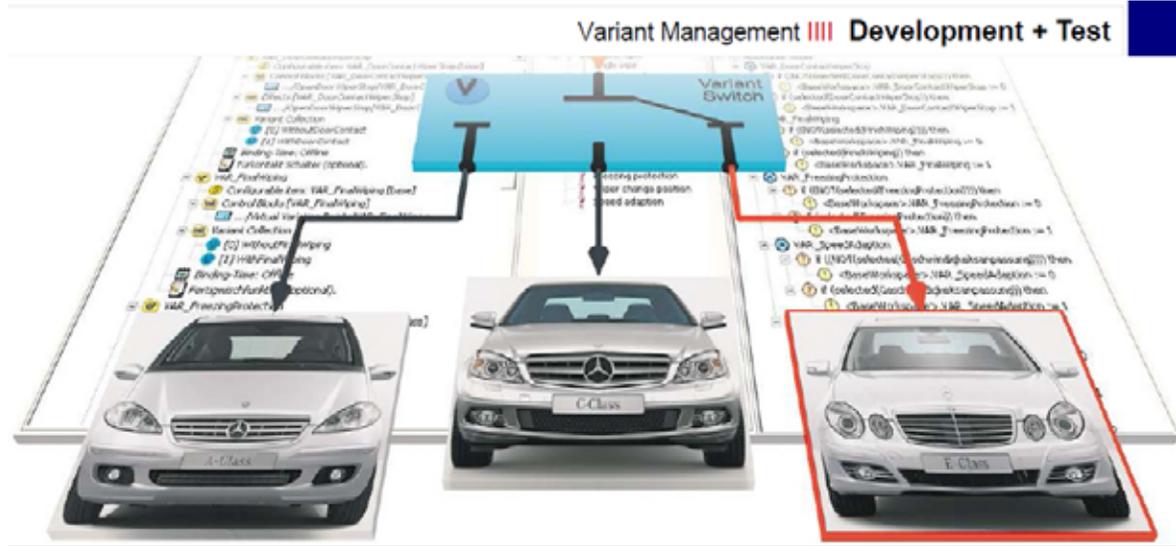


Development + Test III Variant Management



Model Diversity and Variability

モデルの多様性と変動性

Simulinkモデルの機能バリエーションへの対処法 (Handling functional variants in Simulink models)

今日の自動車の特徴は、僅かに異なる機能の広範囲なバリエーションがあることである。これら変動性はソフトウェア開発モデルに反映され、機能モデルの変動性は体系的に扱われる仕組みが必要となっている。主要部分とモデル固有の変動性の情報を区別することで、一貫してSimulinkを扱うようになり、系統だった形式でモデルの変動性を創造できるようになる。

By Christian Dziobek, Joachim Loew, Wojciech Przystas and Jens Weiland

多くの車両機能の存在は、自動車産業界において欠かせない特性となっている。2006年に製造された Mercedes Benz C class の取り得るオプションは広範で、全く同じものが2つとない。その背景は、予想以上に複雑である：異なる地域ごとの法令順守、顧客指向による幅の広い選択肢（ボディのスタイルからエンジン容量まで）殊に、電子機器によってもたらされる様々な機能オプション。これらオプション、機能の大部分はソフトウェアベースであり、組み込みソフトウェアは車両機能の実装の中心的役割を果たしている：車両の機能バリエーション性はソフトウェアのバリエーション性に基づいている。

ソフトウェアのバリエーション性を扱う唯一経済的な方法は、アーキテクチャ記述、システム仕様、ソフトウェア・コンポーネント、ドキュメント、テストデータなどソフトウェア開発の成果物を体系的に再利用することである。これは構造上のバリエーション性を、特定ソフトウェア成果物内の変動性と、組み合わせ上のバリエーション性の両方、あるいは異なる成果物の統合などに対して、極めて系統立った方法で記述することのみ可能となる。

Development + Test III Variant Management

自動車産業界では、組込みソフトウェア開発に、モデルとコードジェネレータの採用が増加している。MathWorks社 Matlab / Simulink / Stateflow などのグラフィカルなモデリング言語は、動的な信号制御と状態遷移ベースのシステムを用いて、デザイン設計、モデリング、シミュレーションが行える。その結果は、MathWorks 社製 Real Time Workshop Embedded Coder や、dSpace 社 TargetLink により実行コードに変換される。Simulink モデルは相互接続可能な基本ブロックとシグナルフローグラフからなる状態図で構築される。複雑なサブファンクションは、サブシステム化され、ブロック間の信号接続がモデルのシミュレーションを介してデータ交換としてシグナルフローグラフに置き換えられる。Simulink 内で、ロジカルオペレータや信号経路設定などの特定機能ごとに基本ブロックの選択は可能になっている。

リアルタイムシステムの機能アルゴリズムを記述するシグナルフローグラフは、基本ブロックを基準にしている。しかし実際には、基本アルゴリズムの再利用可能で、構成可能な機能モジュールは、異なる機能バリエーションのモデリングや構成の必要性和重複し、Simulink では、これらを十分に記述できない。この資料では、Simulink / Stateflow を用いて生成されるシグナルフローグラフにおける、このような記述上の不足を補う取組みに関して紹介する。特に、バリエアビリティを記述する為に基本ブロック内で必要となる情報、およびその情報をどのように保存するかについての考察。

体系的なバリエアビリティに対する考察により以下が可能となる：

- 機能バリエーションの一貫した記述、それに基づく一貫した構成
- シグナルフローグラフ内のバリエーション表記、標準ブロックとバリエーション固有ブロックの識別
- 異なる機能バリエーション間の依存性を認識

ここで紹介される実装には、Matlab 7.1 と TargetLink 2.1 を用いているが、Matlab で提供されるツールチェーンのみでも同様に可能である。

Modelling Variability in Simulink

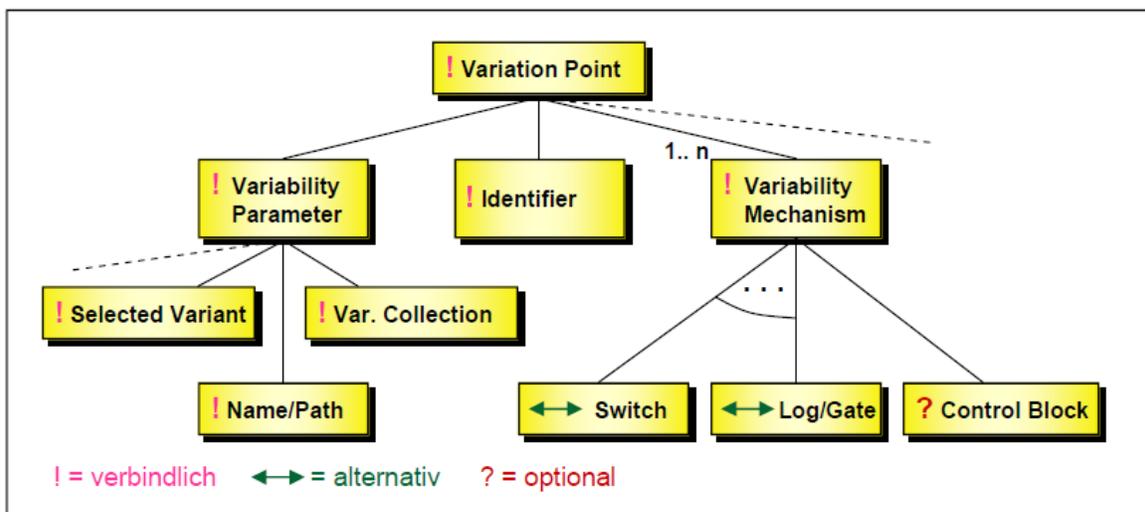


Figure 1: Structure of a Variation Point

Development + Test III Variant Management

アプローチは、Product Family Engineering [1]を基準にし、バリエービリティ記述はバリエーションポイント(変動点)(Figure 1)から行った。バリエーションポイントにより、Simulinkモデルに設定される各機能バリエーションのバリエービリティの情報は包含される。これは、バリエービリティのパラメータとメカニズム(Variability Parameterと、Variability Mechanism)といったユニークな識別子を持つ。

バリエービリティ・パラメータ(Variability Parameter)はSimulinkモデルのチューニングつまみとなり、特定機能バリエーションを選択するために構成される。それは構成されるバリエーションに対する一連の値を持ち、特定の値が選択されるバリエーションに相当する。

バリエービリティ・メカニズム(Variability Mechanism)は、どのように機能バリエーションがSimulinkモデルの特定の部分に実装されるかを記述し、特定のバリエーションがそのバリエーションの構成パラメータに従って実行されることを確実にする。

Simulinkのブロックライブラリは、バリエービリティをモデル化するための多くのブロックを含んでいる。

例えば：

- コンディションにより実行されるサブシステム
(Enabled Subsystem, Function Call Subsystem)
- サブシステム
(IF block, "Switch Case" block)
- 信号経路処理ブロック
(Switch Block, "Multiport Switch" block, "Manual Switch" block)
- ロジックゲート
(AND block, OR block)
- 構造化可能なサブシステム

これらの各ブロックは、可変機能を解決する独自のメカニズムを持つ。Enabled Subsystemは、イネーブル信号により活性化されたり停止されたりするので、オプション機能をモデリングする上で、特に有用である。同様に、オプション機能はロジカルAND、ORブロックでも起動・停止されうる。バリエーションは、Switch Blockのコントロール信号でも選択されうるので、機能選択肢をモデル化することに適している(Cの"Switch Case"文と同様)これらユニットの大部分は、ブロック実行を司る入力信号を必要とする。

バリエーションの選択に、Control Blockを用いることも良い選択肢である：Constant Blockあるいは、"Data Store Read"ブロックとして実装される。これらControl Blockの値により、適切なバリエーションが実行される。

Development + Test III Variant Management

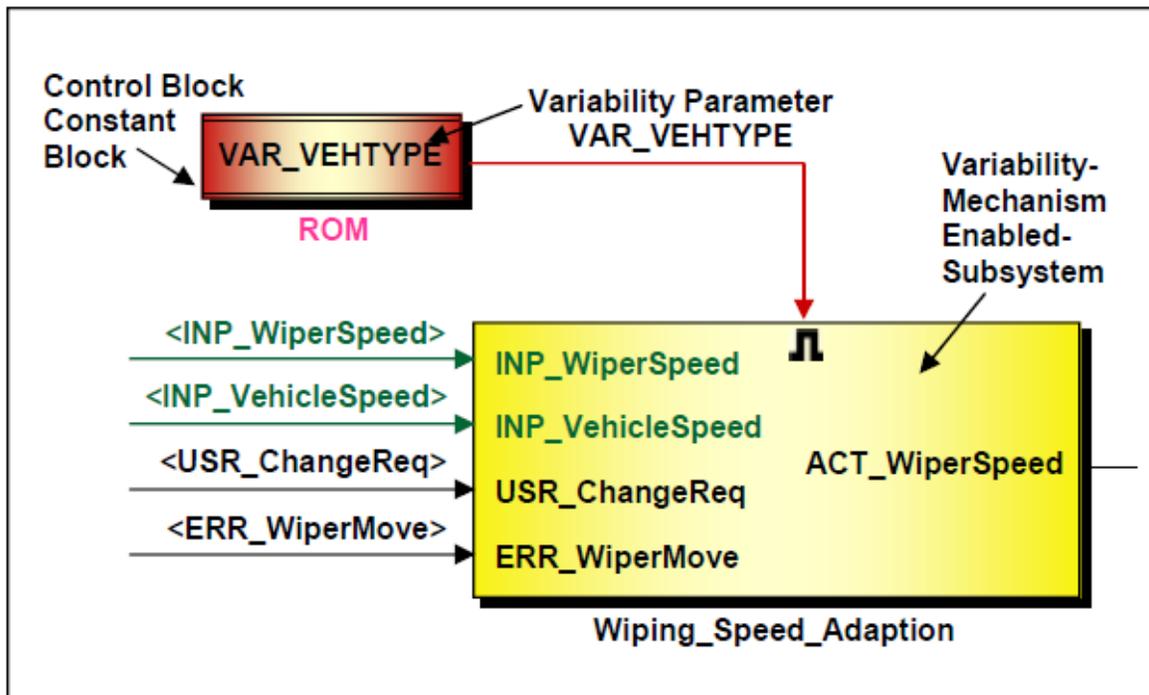


Figure 2: Example of a variation point. Execution is steered via the control block.

Configurable Subsystem は例外と言える：この場合バリエーションは、バインディング・タイムの制限やデバッグ制限、Configurable Subsystem のインターフェイスなどを含んだ Block Parameter Block Choice[2]、を介して選択される。

Figure 2 の Enabled Subsystem は、Constant Block により動作処理が制御されています。この Constant Block は値としてバリエーションを持ち、Block Parameter Value としてバリエーションのパラメータを定義しています。

可変の機能は、モデル上の様々な箇所に影響を及ぼすので、モデル内、あるいはベースとなるワークスペースに中心となるバリエーションのパラメータを生成し、Control Block に参照されるようにすることは、良い方法である：Figure 2 では、中心となるバリエーションのパラメータは、VAR_VEHTYPE

– "Separation of Concerns" 関心事の分離

Simulink 内にバリエーションをモデリングするための情報が明確になれば、異なるバリエーションのメカニズムであっても、統一された手法でバリエーションを取扱い保管されなければならない。Simulink モデルに存在するバリエーションを定義する為に、バリエーションポイント（変動点）を用い、バリエーション・メカニズム（Variability Mechanism）を用いてこれを解決し、バリエーションの情報を、汎用と固有エレメントに分離できるようにする（Figure 3）:

Development + Test III Variant Management

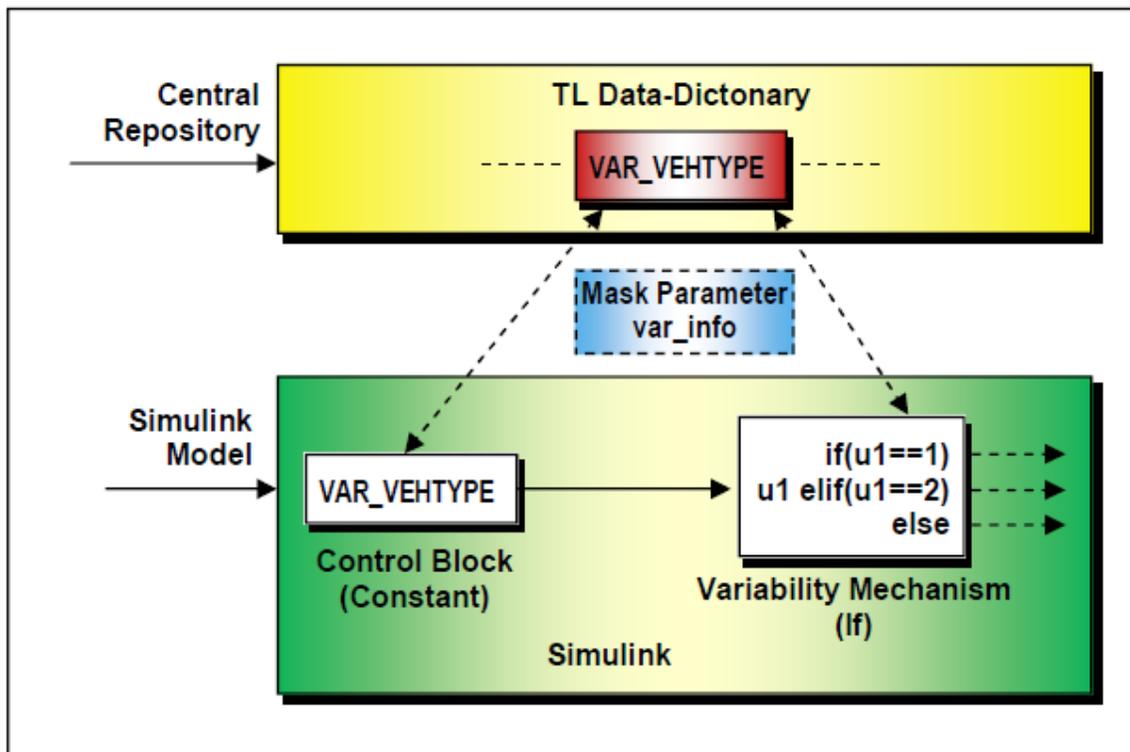


Figure 3: "Separation of Concerns" for Variability Information

- バリエービリティ・パラメータ、バリエーションのコレクション、選択されたバリエーションなどが一般的なバリエービリティ情報を象徴し、同一タイプのオブジェクトとして中心となるバリエーションデータベースに保管される。各オブジェクトはユニークなバリエーションポイントといった識別子を持ち、異なるコントロール・ブロックや、モデルのバリエービリティのメカニズムから参照される：この例では、TargetLink Data Dictionary 内で、データはあるオブジェクトとしてモデル化されている。あるいは、Matlab Structure、Simulink Data Class、またはカプセル化された Java Class としてインスタンス化され、ベースモデルやワークスペース内のパラメータとなる。

- バリエービリティ・メカニズム、それに関連するコントロール・ブロックにより特定の元素が供給される。Simulink モデルのバリエービリティを表現するブロックは、追加のマスクパラメータ VAR_INFO でバリエーション固有に定義され、その値は中心となるバリエーションデータベース内の相当するオブジェクトを指示す。Tag や Description などのブロックパラメータが既に他の目的で使用されているので、マスクパラメータの使用は、ユニークな識別子をバリエービリティ情報として割り当てることである。

Development + Test III Variant Management

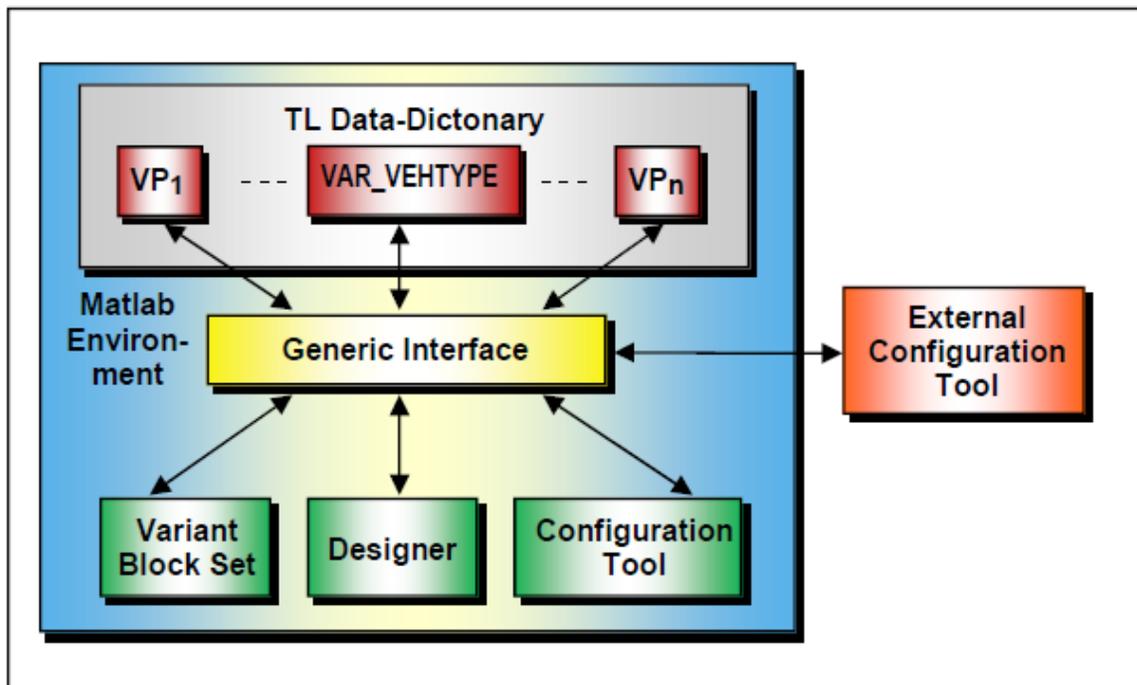


Figure 4: Generic interface for transparent access to Variability information

バリエーシティ情報を共通部分と特定パーツに分割することで、バリエーシティ情報に対して、汎用のインターフェイスを介して中心となるバリエーションデータベースへの一貫したアクセスを提供する。(Figure 4)

このインターフェイスで、変動機能を解決する為の独自のメカニズムを持つ、異なるブロックは識別可能なバリエーションポイント構造を用いながら、バリエーシティ・メカニズム (Variability Mechanism) として使用される。そして設計者はSimulink モデル内のバリエーシティに対し、Matlab ファンクション、設定ツールを介して、明白なアクセスが可能になる。

バリエーシティをモデリングする体系的なアプローチを開発していく一環として、分離されたバリエーションブロックセットをデザインした。

それは特別に注釈付きの汎用Simulink ブロックのライブラリで、"Callback" パラメータ OpenFcn により、バリエーション固有のダイアログを用いて設定可能なブロックである。

Model-based Development Process

どのようにバリエーション・ブロックセットを用いてバリエーシティをモデリングするかの例を示す(Figure 5) 始めにバリエーション固有のブロックがSimulink に挿入される(Figure 5, 左下)。これらは後程、Variant Specific Dialogue を用いて構成される(Figure 5, 右下)。これは"Open Callback" 機能Open-Fcnを用いて呼び出され、既存のバリエーションポイントがバリエーションデータベースに保管され、編集や、新しいバリエーションポイントの生成が出来るようになる。

Development + Test III Variant Management

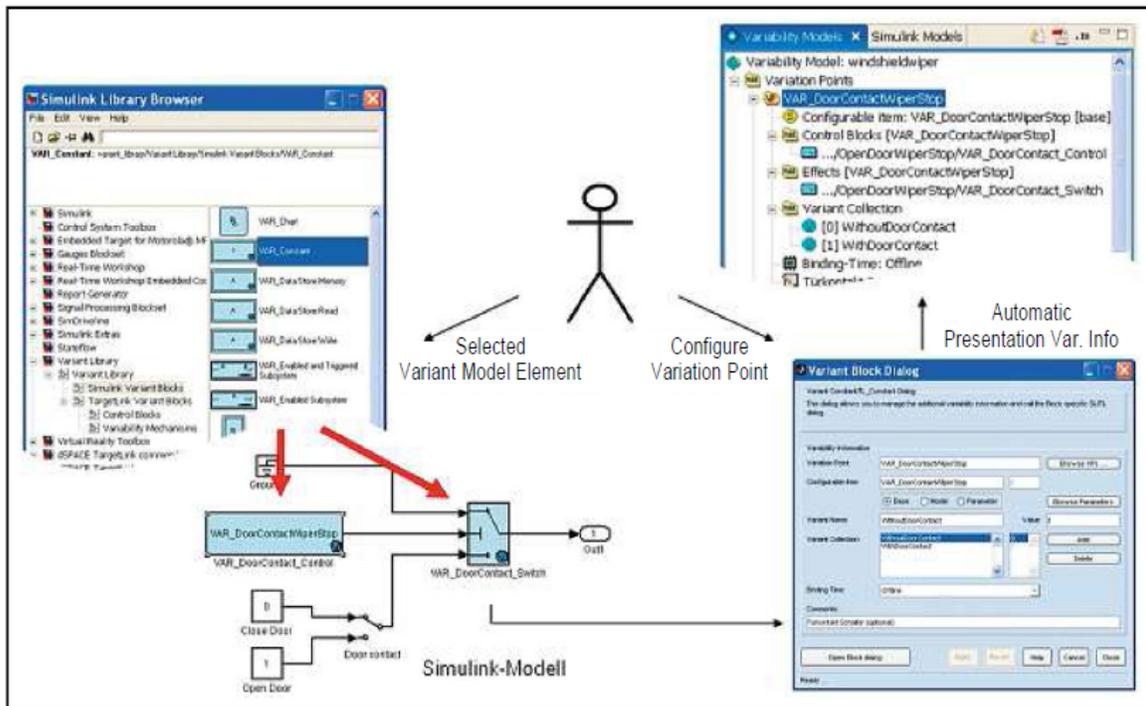


Figure 5: Design of variable functions based on variant block sets

新しいバリエーションポイントを規定する時に、バライアビリティのパラメータは以下によって定義される：

- 名前とパス
- パラメータが取り得る値
- 選択されたバリエーション

Variant Specific Dialogue は、TargetLink のブロック使用時に、ブロック固有の情報を入力することもできるようになっている。Simulink モデルはツリー構造で一覧表示があり (Figure 5, 右上)、これは明白なバライアビリティの表現として、入手可能な情報源として自動的に得ることが出来る。バリエーション固有のブロックを選択することで、Simulink モデル内の相当するブロックが直接呼び出される。

Development + Test III Variant Management

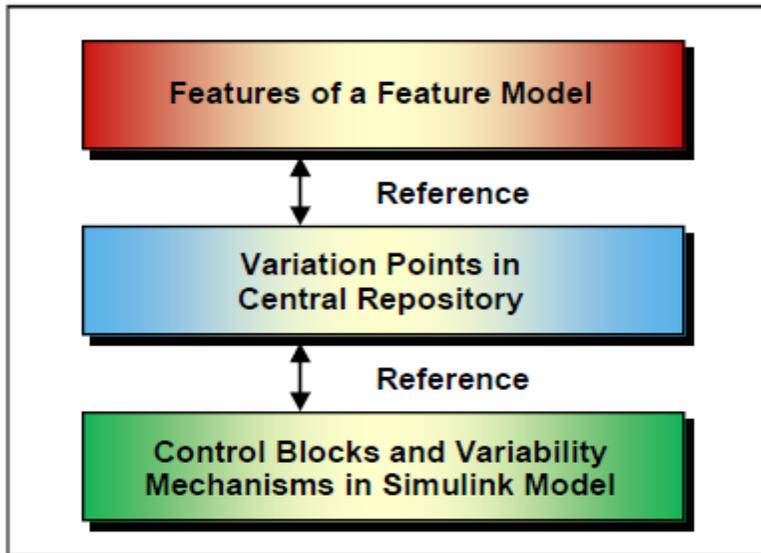


Figure 6: Levels of Abstraction from Feature to Simulink Model

このような、Simulink モデル内のバリエビリティへの明白なアクセスは、広範なオプションを構成ツールに提供する。 Feature Models を用いるそのようなツールは、機能バリエーションに対して抽象的な視点を提供し、Simulink モデルとは独立している (Figure 6)。

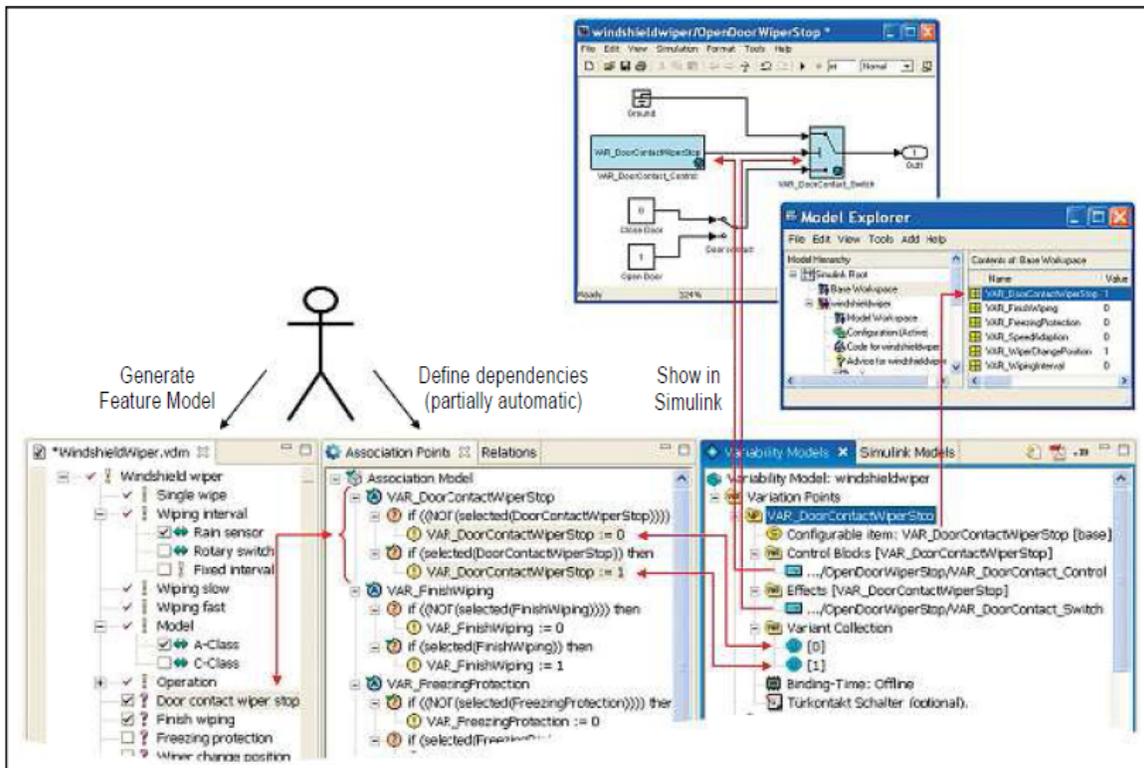


Figure 7: Configuration of Variability in Simulink Models via Feature Models using Door Contact information

Development + Test III Variant Management

フィーチャーは、ユーザの観点からドメイン・スペシフィックの共通部分と変動要素を規定する。(Figure 7 の例では、door-contact 機能オプションを選択すると、フロントドアが開かれる時にワイパーが停止する) 結果、これらコンセプトの包括的なモデルは、モデルバリエーションのコンセプトの依存関係を含み(Figure 7, 左下)、必須、選択肢、任意機能などの選択肢、(1..n):m のグループ相関、などを識別できる。 Feature Model のフィーチャと、Simulink モデルのバリエーションポイント間の関係、およびそれらがどのように定義されるか(in Figure 7, 真ん中下の、Association Model)、などはReference [3] で解説されている。

これらの関係から、Simulink モデル内のバリエーション固有のパラメータ値を、Feature Model 上の特定フィーチャに割当てることが可能で: pure-systems 社の pure::variants ツールは、このアプローチをサポートするために、Simulink モデルと統合済である [4]。

要約すると、このアプローチでは重要なフィーチャ・バリエーションへの対処を支援し、モデルベース・ソフトウェアの品質向上に役立てられる。

* 依存関係を定義し、Simulink モデル内に展開されたバリエーションを選択的に探索可能。 Simulink モデルのバリエーション・ポイントに影響するフィーチャと、バリエーション・ポイントによって決まるフィーチャの両方を探索して。

* Feature Model から、自動的に正しいSimulink の構成を生成させることが可能。例えば、パラメータセットや、構成仕様書など。

* Feature Model とSimulink モデルの連結により、複雑なバリエーションの機能モデルの、構成上の矛盾を探索可能。

* Simulink モデル内のバリエーションは、明示的に視覚化される。 全てのバリエーションは、Simulink モデル内のバリエーション固有ブロックと共通のバリエーションデータベースで表され、Simulink モデルの全バリエーションは、集中管理され得る。

ここで紹介したコンセプトは研究プロジェクトとして開発され、現在 Mercedes 社の自動車製品開発に採用されている。特に、専用の Variant Block セット、Matlab ファンクションに基づいた API 、そして構成ツールをこの目標を実現させる為に開発した。

Literature: Original

[1] Czarnecki, K.; Eisenecker.

U.: Generative Programming – Methods, Tools, and Applications. Addison-Wesley, Boston, MA, 2000.

[2] Weiland, J.; Richter, E.: Konfigurationsmanagement variantenreicher

Simulink-Modelle. A.B. Cremers et.al. (Hrsg.): INFORMATIK 2005 – Informatik LIVE!. Band 2. Beiträge der 35. Jahrestagung der Gesellschaft für Informatik e.V. (GI). 19. – 22. September 2005. Kölln Verlag, Bonn, 2005.

[3] Klengel, K.; Weiland, J.: Merkmalbasierte Konfiguration variantenreicher Simulink- Modelle. H. Dörr, T.

Klein: Unterlagen zum Workshop „Modellbasierte Entwicklung von eingebetteten Fahrzeugfunktionen“, Modellierung 2006. 22. – 24. März 2006. Innsbruck, Österreich, 2006.

[4] pure-systems GmbH, pure::variants. Eclipse Plugin User Guide, 2007.

Development + Test III Variant Management



**Dipl.-Ing.
Christian Dziobek**

studied Electrical Engineering at RWTH Aachen. He has worked at Daimler AG since 1998, specifically on the introduction of methods and tools for model-based function development at department E/E for series car development.
christian.dziobek@daimler.com



**Dipl.-Inf.
Wojciech Przytas**

studied Computer Science and Computer Linguistics at Stuttgart University. He became a doctoral student at Daimler-AG-Research in 2007 and works mainly on variant configuration of model-based embedded software.
wojciech.przystas@daimler.com



**Dipl.-Ing.
Joachim Loew**

studied Aerospace Engineering at Stuttgart University. He has worked at Daimler AG since 1998. His responsibilities include the development of model-based function design tools for ECUs.
joachim.c.loew@daimler.com



**Dipl.-Inf.
Jens Weiland**

studied Computer Science at Bundeswehr University in Munich. Since 2000 he has headed research projects at Daimler-AG Research. His focal point is variant configuration of model-based embedded software.
jens.weiland@daimler.com