



Highlights in pure::variants Release 4.0.19

Release 4.0.19のハイライト

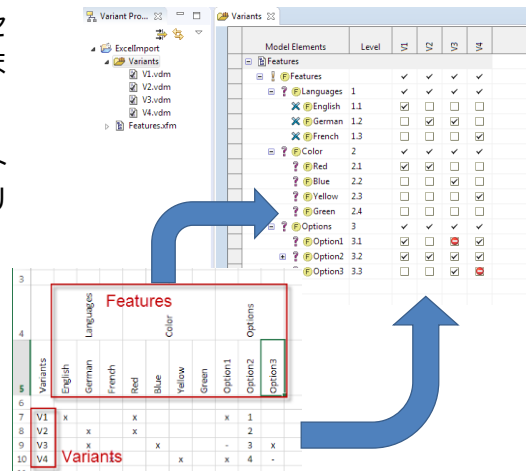
© pure-systems GmbH

pure::variants の最新リリースで強化された内容をご紹介します。

汎用のエクセルインポート

- フィーチャとバリエントをエクセルの「名前の定義」でマークします
- そのエクセルデータをインポートすると、フィーチャモデルとバリエントモデルが生成されます

pure::variants User's Guide 4.0.19
のp.99をご覧ください



© pure-systems GmbH

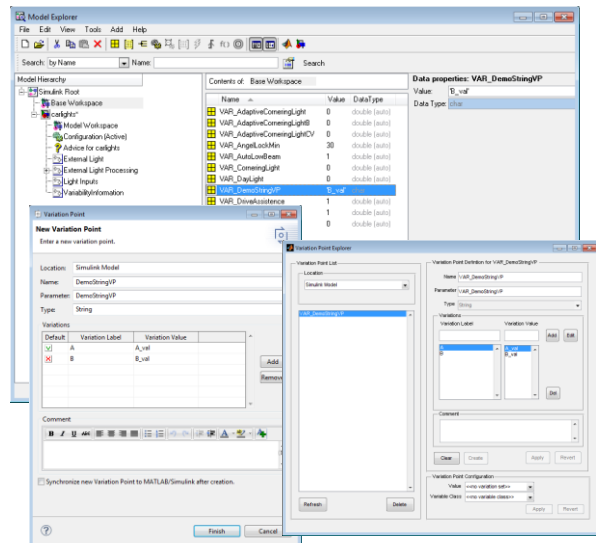
国内のお客様から、
エクセルでバリエント管理を試みたが、制限があることに気が付いたので
pure::variantsに切り替えたい。
ただフィーチャモデルやバリエントの定義を一から作るのではなく、今まで
エクセルで記述してきたものをインポートしたい。
という要望がありました。

以前からCSVファイル形式をフィーチャモデルにインポートできていましたが、
最新版ではバリエントモデルにも対応しました。

たとえば表の列がフィーチャ、行がバリエントと指定して、バリエントごとの
フィーチャの選択情報も合わせてpure::variantsにインポートできるようになり
ました。

Simulinkのバリエーションのパラメータにstringをサポート

- Simulinkのバリエーションポイントのパラメータが、数値に加えて、stringデータも使えるようになりました
- pure::variantsのモデル構成の設定などでそれを利用することができます
- 注): stringによるバリエーションは、データ辞書内ではなく、モデル内に保存されます



© pure-systems GmbH

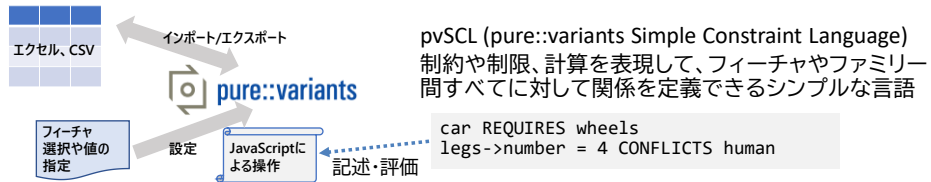
これも国内のお客様の要求に応じて強化されました。以前のバージョンでは、Simulinkのバリエーションのパラメータ（Variation Value）は数値のみでしたが、今回のバージョンから、stringデータが扱えるようになりました。そのため、各バリエーションのパラメータにライブラリリンクを設定することで、バリエーションの選択に応じて、ライブラリモデルを切り替えることができるようになりました。

（バリエーションの選択に応じて、これがSimulink内のパラメータとして設定されて、ライブラリモデルを切り替えることができるようになりました）

独自ツールへの対応: JavaScript 変換での pvSCL 演算

- JavaScript 変換の中で、pvSCL 文のオンザフライ評価が容易にできます

- `var calcResult = Evaluator.calculation("FeatureA->speed / 100");`
- `var ruleResult = Evaluator.rule("FeatureA OR NOT(FeatureB)");`



- これにより、カスタムデータに pvSCL 文を埋め込んで、そのデータ資産に対する変換を JavaScript で完全に記述することができます

pure::variants ブログ <https://pure-variants.blogspot.com/> をご覧ください

© pure-systems GmbH

これは簡単に言うと、pure::variants にサポートされていないユーザ独自のツールやデータでも、対応させることが容易になったということです。

pure::variants には JavaScript インターフェースが提供されており、JavaScript で記述して外部に定義したフィーチャをインポートすることや、バリエーションモデルを操作したりできます。

また、フィーチャの制限や制約を記述する言語 pvSCL があります。

最新版から、pvSCL を JavaScript で直接記述して評価できるようにしました。これによりサポートされていないユーザ独自のツールやデータへの対応が、ユーザ側で容易に行えるようになりました。

Desktop Hub から JavaScript の編集などができますので、pure::variants と統合されていないツールに対しても pvSCL を使用した JavaScript 変換ができます。

(先にご紹介したエクセルインポートも、これで実装されています)

pvSCLの新しい関数

- 追加された数学関数

- Abs, Exp, Log, Log10, Max, Min, Mod, Pow, Sqrt, Sum, Sin, Cos, Tan, Asin, Acos, Atan

- 例: `pv:Log10(TestFeature->value) > 1`

- コレクション用の新しい関数

- Insert, InsertAll, SubList, Sort

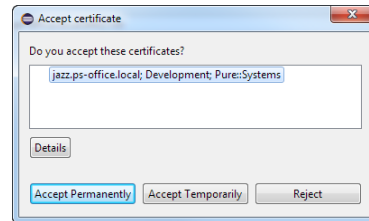
- リストのソートでの使用例: `{1,3,2}->pv:Sort()` は `{1,2,3}` を返します

pure::variants User's Guide 4.0.19の9.8.21節 Function Library、p.188以降をご覧ください

pvSCL自体にも機能追加として、新しい関数が追加されています。
対数関数のような数学関数や、ソートのようにリストなどを操作する関数を
例として挙げています。

自己署名証明書「オレオレ証明書」の扱いを改善

- pure::variantsとすべての統合ツールでは、ライセンスサーバー、pure::variantsサーバー、またはツール統合による通信の暗号化で使用される証明書が、自己署名であるか、信頼されないか、期限切れか、を承認/拒否するようユーザーに要求します
- その決定を一時的(Accept Temporarily)か、または永続的(Accept Permanently)に保存します



© pure-systems GmbH

このスライドは、自己署名証明書「オレオレ証明書」の扱いが、より簡単にできるように改善したという内容です。

* 「自己署名証明書」は自分で自分の正当性を証明するという構造になっていますので、オレオレ詐欺のように、信頼できる組織を騙ることができてしまい、このことを指して「自己署名証明書」を信頼できない証明書という意味で俗称として「オレオレ証明書」と呼ぶ場合もあるほどです。「自己署名証明書」であってもSSL通信自体は暗号で行われるので、悪意ある第三者による「盗み見」は防がれています。しかし悪意のある第三者が接続先サーバを偽装して「フィッシング詐欺」などを行うことが可能となってしまいます。

pure::variants サーバーへのシングルサインオン

- 連携する外部システムを一度の認証で利用できる
- pure::variantsサーバーのプロジェクトに対してのみ適用可能
 - サーバー統合認証とLDAPに加え、pure::variantsでは以下を容易に利用できます
 - Windows 認証
 - Open ID Connect

pure::variants Database Server Install Manualの p.8をご覧ください

© pure-systems GmbH

またシングルサインオンで使えるようにもなりました。

これらはバリエーション機能の改善ではありませんが、利便性を向上しつつ、セキュリティリスクを軽減することに役立つかと思えます。

このような強化がされたということから、近年pure::variants が実践的に活用されるようになってきたことを示すことにもなるかと考えております。

* **Single Sign On** は、1つのIDとパスワードを入力して、複数のWebサービスやアプリケーションにログインする仕組み。入力や管理の手間を省き、セキュリティを強化することができる。例えば、メール、SNS、Google、Amazon、グループウェアなどのサービスごとに別々のIDとパスワードを設定すると、アクセスするたびにパスワードを入力する労力がかかる上、パスワードを忘れると再設定する手間もかかる。また、簡単なパスワードを設定して使い回したり、メモなどで残したりすると、流出や漏えいのリスクが高くなる。シングルサインオンは1つのIDとパスワードでサービスが利用できるため、パスワードを管理する負担から解放され、利便性が向上し、セキュリティリスクが減る。また、企業のシステム管理者の労力も省かれ、生産性が向上する。