

製品間を横断した ソフトウェア共通化技術

～ソフトウェアプロダクトラインの最新動向～

吉村健太郎
Hitachi Europe

携帯電話、デジタルカメラ、自動車等、我々の身近な電子製品において、組み込みソフトウェアは製品の競争力を左右する重要な要素である。ところが、製品機能の増加に伴ってソフトウェアは大規模・複雑化しているにもかかわらず、1機種あたりの開発期間は大幅な短期化が求められている。

この問題を解決するためには、組み込みソフトウェアを機種間で共通化・再利用化することで、機種あたりの開発コストを低減することがきわめて重要である。本稿では、類似製品間でのソフトウェア共通化技術として産業界からも注目されているソフトウェア・プロダクトライン工学（SPLE：Software Product Line Engineering）の概要、最新の適用事例および研究トピックについて述べる。

組み込みソフトウェアの多品種化と再利用

携帯電話、カメラ、自動車等、我々の身近な製品のエレクトロニクス化が急速に進んでいる。それに伴い、組み込みソフトウェアの多品種化、大規模化および高信頼化が課題になっている。たとえば自動車を例にとると、初期の製品はほぼ機械部品で構成されていたといえる。現在の自動車は、ハイブリッドシステムや車間距離制御システム、カーナビゲーションシステムなどさまざまな電子システムを搭載しており、新機能の90%以上は組み込みソフトウェアを用いて実現されている。

これら組み込みソフトウェアは、当初は単一の製品向けに開発されることが多い。その後、製品の成功とともに改良、派生が加えられ、複数の製品系列として開発されるようになる。たとえば、エンジン制御部品メーカーは、複数の自動車メーカー向けにそれぞれ複数のエンジン制御システムを開発している。これらのエンジン制御システムは技術的には共通のバックグラウンドを持っているが、エンジン仕様の違いや、自動車メーカーごとの独自要求仕様、さらに地域ごとに異なる法規制に対応するため、組み込みソフトウェアのバリエーションは非常に多い。さらに、組み込みソフトウェアではハードウェア・コスト低減のために処理性能やメモリサイズに制限があり、実装されるソフトウェアは各製品向けに最適化されたものとなる。

ソフトウェア規模・複雑度の増大、開発コスト・期間の削減、信頼性の向上等の要求により、製品バリエーションを個別に開発することはきわめて困難である。そのため、組み込みソフトウェアの再利用性向上が重要な課題になっている。組み込みソフトウェアの再利用性向上に

より、以下の効果が期待できる。

- 既存ソフトウェアの再利用による開発工数の削減
- 検証済みソフトウェアの再利用による品質保証
- 製品個別機能開発への工数集中による多品種開発

近年、製品系列を横断したソフトウェア再利用技術としてソフトウェア・プロダクトライン工学（以下、SPLE：Software Product Line Engineering）が注目されている。本稿では以下、SPLEの概要、適用事例、研究トピックおよび今後の展望を解説する。

SPLEの概要

◆ 研究の流れ

SPLEの研究の流れは、大きく2つに分けられる。1つは欧州での「Product Family Engineering (PFE)¹⁾」、もう1つは米国での「Software Product Line (SPL)²⁾」である。

PFEは欧州連合の研究プロジェクトARES^{☆1}により研究が開始され、1996年に初めてのワークショップが開催された。SPLEの研究は後継プロジェクトであるEsaps(1999-2001)、Café(2001-2003)、Famillies(2003-2005)に引き継がれ、Philips, Nokia, Siemens, Thales, Telvent等の欧州企業と、大学、研究機関によって、研究および事例適用が進められてきた。

SPLはCarnegie Mellon University, Software Engineering Institute (CMU/SEI)により提唱された概念で、HP社、Cummins社、Motorola社等のソフトウェア再利用の事例を収集・整理したものである。CMU/SEIは2000年からSPLに関する国際会議(SPLC：Software

☆1 Architectural Reasoning for Embedded Systems

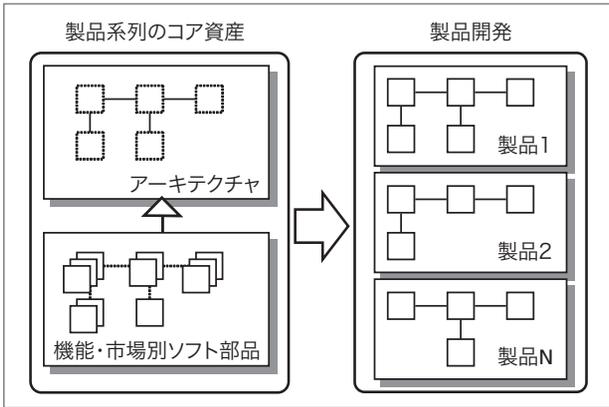


図-1 SPLEによる開発

Product Line Conference) を開催してきた。

近年では、「ソフトウェア・プロダクトライン工学 (SPLE)」をひとつの技術体系として捉えようという動きが活発になり、SPLCは2005年から欧米共催の国際会議として開催されている。2006年10月開催のSPLC2006には、22カ国から190人以上が参加した。SPLEは日本、韓国でも注目を集めており、来年2007年9月には京都でのSPLC開催が決定している^{☆2}。また、業界誌でも特集記事としてSPLEが取り上げられるなど、産業界においても注目が高まっている³⁾。

◆ SPLEの基本概念

図-1に、SPLEによる開発の例を示す。SPLEの骨格は、製品系列を横断する共通的なプロダクトライン・アーキテクチャと、それに基づく再利用資産の体系化である。効率よく再利用可能なコア資産を開発するためには、類似する製品間の共通性 (commonality) はもちろん、可変性 (variability) を管理することが重要となる。その上で、プロダクトライン・アーキテクチャを設計し、コア資産を整備する。そして製品個別の仕様を決定したときに、コア資産を利用し、かつ製品固有の機能を開発して、最終的な製品を開発する。

SPLEの特徴は、コンポーネント指向やフレームワークといった技術的な視点に加え、企業としてのビジネスの視点を積極的に取り入れた点にある。欧州で開発されたSPLEではBAPO (Business, Architecture, Process and Organization) と呼ばれる概念がその基盤となっている¹⁾。この4つの概念は相互に影響しており、どれか1つが欠けても再利用は成功しないといえる。図-2は、BAPOの各要素が互に関連していることを示している。

● ビジネス (Business)

製品系列を開発する際には、まず始めに事業として利

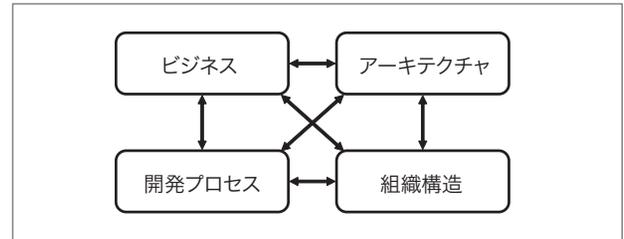


図-2 BAPO

益を挙げ得る製品系列の範囲 (scope) を決定する必要がある。製品系列の範囲が広すぎれば、コア資産の開発に多大なコストを要する。逆に製品系列の範囲が狭すぎれば、市場の要求に応じた製品開発が難しくなる。この範囲の決定には、当該製品分野における将来の市場動向分析と製品計画という、ビジネスの視点での検討が必要不可欠である。製品系列の範囲を適切に決定することによって、上級機種と普及機種間での再利用、現行製品と将来製品での再利用が可能になり、開発工数の削減が可能となる。

● アーキテクチャ (Architecture)

製品計画に基づいてプロダクトライン・アーキテクチャを設計することによって、マーケット領域をカバーするソフトウェアを効率よく開発できるようになる。プロダクトライン・アーキテクチャは、単一製品のアーキテクチャではなく、製品系列全体を対象とする。そのため、製品系列内での共通性と可変性を表現している必要がある。このアーキテクチャに基づき、製品系列内で利用するコンポーネントを開発・実装する。個別の製品は、共通性・可変性に対応する開発・検証済みのコンポーネントと、製品固有のコンポーネントを組み合わせて開発されるため、信頼性の高い製品を効率よく開発することができる。

● 開発プロセス (Process)

製品系列の開発は大規模な組織で行われることが多いため、プロセス定義は必要不可欠である。SPLEの活動では大きく分けて、従来からの製品開発 (Application Engineering) のプロセスに加えて、製品系列で共通に利用するコア資産開発 (Domain Engineering) のプロセスが定義されている。

● 組織構造 (Organization)

さらに、以上の活動を円滑に行うための組織構造も欠かせない。組織構造は個別製品に対応して構成されていることが多いが、製品系列を横断したコア資産開発の役割をどのように割り当てるかが重要なポイントとなる。従来、製品別に開発していた機能をコア資産として共通に開発・利用するため、開発工数を製品固有機能に投入することができ、多品種開発が可能となる。

☆2 <http://sec.ipa.go.jp/SPLC2007/>

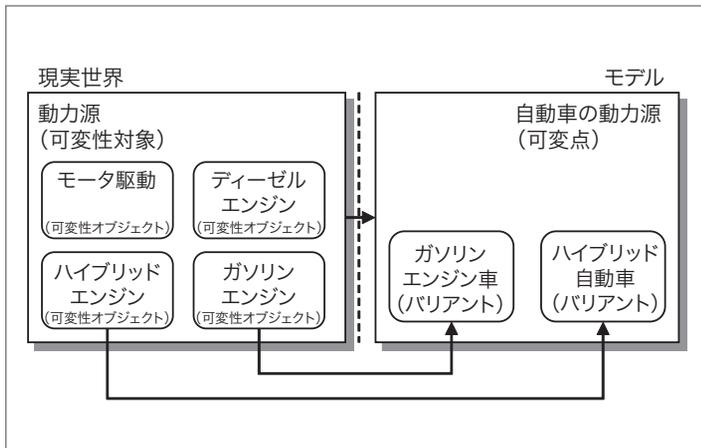


図-3 可変性の例

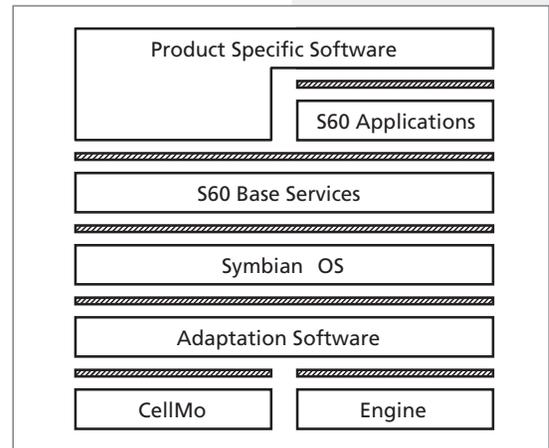


図-4 Nokia 社 S60 プラットフォーム

◆ 可変性

可変性とは、製品系列内での製品間において「どこが」「どのように」異なっているかということに関する想定である。Pohlら⁴⁾の定義によれば、可変性は、物理的な実装または製品の特性（フィーチャ）が変化する可変性対象（variability subject）と、可変対象の具体的な実現である可変性オブジェクト（variability object）からなる。さらに、可変性対象をコア資産内に設計情報として表現したものを可変点（variation point）と呼び、可変性オブジェクトのコア資産内での実装がバリエーション（variant）となる。

図-3に可変性の具体例として、自動車の動力源の例を示す。この図の例では、図中右側の「自動車の動力源」が可変点となる。そして、図中左側の「動力源」が可変性対象であり、可変点「自動車の動力源」に対応する。さらに、可変性対象「動力源」に対応する具体的な実現である可変性オブジェクトは、ガソリンエンジン、ディーゼルエンジンおよびハイブリッドエンジン等、複数想定される。

自動車メーカーがガソリンエンジン車およびハイブリッド自動車を製造する場合には、図中右側に「ガソリンエンジン車」および「ハイブリッド自動車」がバリエーションとして定義される。この例では、ディーゼルエンジンやモータ駆動など、他の可変性オブジェクトは、マーケットの傾向やコスト、技術的な問題から、バリエーションとしては選択されていない。

適用事例

本章では、SPLEの実製品への適用事例とその効果を紹介する。

• Cummins 社

同社では、ディーゼル・エンジン制御ソフトウェアに

SPLEを適用し、1,000種以上の製品バリエーションをプロダクトラインから開発している。同社では約100名のスタッフによりソフトウェアを開発しているが、SPLEを適用しなかった場合には360名以上のスタッフが必要であると見積もっており、SPLEによる生産性向上は3.6倍となる。

• Nokia 社

同社では、年間50種以上もの携帯電話をリリースしている。多品種開発を実現するため、Nokia社はローエンド向けのS40、ミドル・ハイエンド向けのS60、スマート・フォン向けのMAEMOプラットフォームを採用している。製品機能に占めるプラットフォームの利用率は、S40では100%、S60では60%、MAEMOでは30%である。図-4にS60プラットフォームの概要を示す。

• MARKET MAKER Software 社

同社は株式情報表示用ソフトウェアの開発を行っている。インターネット向けサービスの開発において、異なるプラットフォーム、異なるデータベース、顧客の要求、表示フォーマット等に適用する必要があり、SPLEを導入した。その結果、開発期間を50%短縮し、開発コストを70%削減した。

• General Motors 社

同社では、自動車用パワートレイン制御ソフトウェアにSPLEを適用している。同社のプロダクトラインは5種類のエンジン制御装置、4種類の変速機制御装置および3種類のパワートレイン制御装置をサポートしている。また、パワートレインの物理的な構成（ガソリンエンジン or ディーゼルエンジン）のバリエーションもサポートしており、結果として100車種以上の製品をプロダクトラインから開発している。同社では、ガソリンエンジン制御ソフトウェアのバリエーションを17種から3種に削減できると予想している。

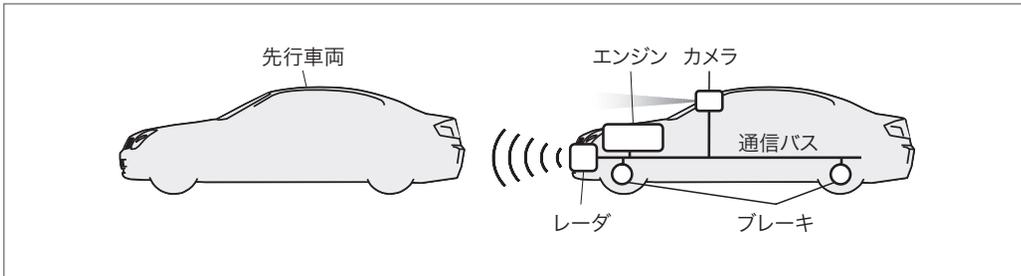


図-5 クルーズ制御システム

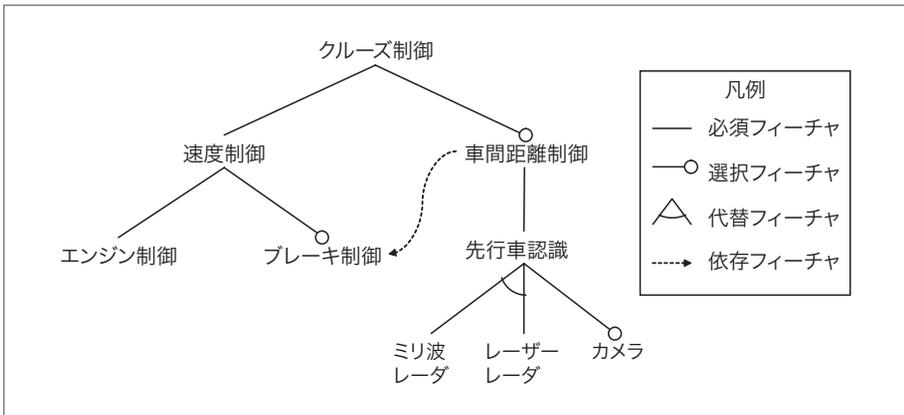


図-6 Kangらのフィーチャモデルによる可変性のモデル化

● Lucent 社

同社では、電話交換機制御ソフトウェアに対して SPLE を適用し、開発効率を 3～5 倍程度向上させた。以上のように、特に多品種のソフトウェアを開発する必要がある組織において、SPLE は大きな効果を挙げている。しかしながら、SPLE の研究の歴史はまだ浅く、多くの課題が残されている。次章では、現在活発に研究が行われているいくつかのトピックを紹介する。

研究トピック

本章では、SPLE に関して現在活発に議論が行われているトピックについて紹介する。

● 可変性のモデル化

可変性は SPLE において非常に重要なポイントである。複数の製品を「製品系列」として扱うためには、製品間の共通点はもちろん、違いを明確に分析し設計に取り込む必要がある。可変性のモデル化方法や、可変点間の依存性の記述、そして要求、設計、実装、テストへのマッピング・管理手法について多くの議論がされ、複数のモデル化手法が提案されている。

● プロセス管理・経済モデル

組織的な再利用活動は SPLE の大きな特徴の 1 つであり、再利用を促進するための開発プロセスの研究が多く行われている。

● 既存システムからの移行

現実問題として、「製品系列」と呼べるほどの製品数

がある場合には、すでにソフトウェアは開発済みであり、かつ SPLE の概念には基づいていないことが多い。そのため、開発済みの既存システムを SPLE 型開発に移行するためのリエンジニアリング手法が提案されている。

以下、各トピックについて詳細に解説する。

◆ 可変性のモデル化

本節では、具体例を用いて可変性のモデル化の例を紹介する。具体例として、図-5 に示す、クルーズ制御システムを用いる。本システムは、自動車の走行速度を設定車速に保つ走行支援システムの一つである。また、本システムの応用システムに、先行車両に追従走行する車間距離制御システムがある。車間距離制御システムの実現には、レーダシステムやブレーキ制御機能が必要となる。レーダシステムには、ミリ波レーダおよびレーザーレーダがバリエーションとして存在する。

Kang らによって、製品系列の機能および品質特性（フィーチャ (feature)）に着目して可変性の分析、モデリングを行う手法が提案されている⁵⁾。フィーチャモデルは、製品系列に含まれるフィーチャを階層的に分析した上で、共通フィーチャと可変フィーチャとを分類、定義する。可変フィーチャには、選択的 (optional) フィーチャと代替的 (alternative) フィーチャが含まれる。また、可変フィーチャ間の依存性、排他性、相反性などの関係も定義される。

図-6 に、フィーチャモデルによるクルーズ制御シ

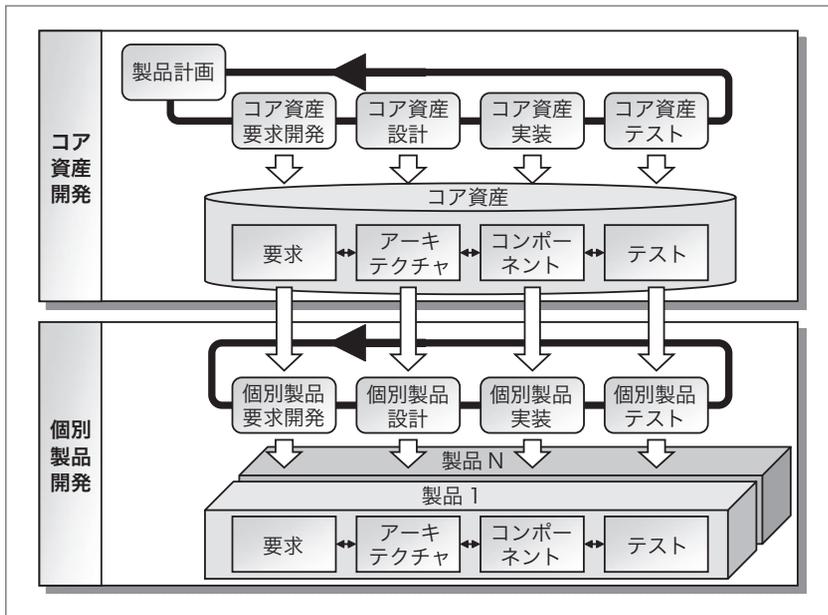


図-7 SPLEの開発プロセス

$$ROI = \text{Cost savings} / \text{Cost of investment}$$

$$= \frac{\sum_{i=1}^n C_{evo_i} - \left(C_{org} + C_{cab} + \sum_{i=1}^n (C_{unique}(p_i) + C_{reuse}(p_i)) \right)}{C_{org} + C_{cab}}$$

C_{evo} - 非SPLE開発による製品あたり開発コスト
 C_{org} - 組織コスト(教育, 訓練)
 C_{cab} - コア資産開発コスト
 C_{unique} - 製品固有部分開発コスト
 C_{reuse} - コア資産再利用コスト
 n - 製品系列に含まれる製品数

図-8 SPLEのROI計算モデル

テムの記述例を示す。この例では、車間距離制御フィーチャが選択的なフィーチャであり、本システムが実装される車種によって選択される。また、先行車認識フィーチャ下部では、代替的なフィーチャとしてミリ波レーダまたはレーザーレーダのいずれかが選択される。そして依存関係として、車間距離制御フィーチャにはブレーキ制御フィーチャが必要なことを示している。

また、上記のフィーチャモデルは独自の記述方法をとっており、UMLで記述された従来の設計資産との関連性を上手く記述できないため、UMLのクラス図を拡張した可変性のモデル化手法がGomaaなどにより提案されている⁶⁾。またPohlらは、可変性は要求分析、設計、実装およびテストの分析結果に対して直交する概念であり、従来のモデル化方法では十分に記述できないとの立場から、可変性を明示的に表現し、影響を追跡

するための直交可変性図 (Orthogonal Variability Modeling) を提案している⁴⁾。

◆ プロセス管理・経済モデル

図-7にSPLEの開発プロセスの一例を示す⁴⁾。図の上段はコア資産を開発する段階、図の下段はコア資産に基づいて個別の製品を開発する段階である。それぞれの段階における活動の定義や、活動間での一貫性の管理方法について研究がなされている。

• コア資産開発

製品開発に再利用可能なコア資産の開発を実施する。コア資産は、要求、設計、実装、テスト等あらゆる種類の開発資産を含んでいる。また、体系的かつ一貫した再利用のため、コア資産間のトレーサビリティ管理も必要不可欠である。

• 個別製品開発

この段階では、コア資産に基づいて個別の製品を実現する。また、この段階で新たに開発される機能は、コア資産からの差分で与えられる。開発済みのコア資産と製品固有の機能とを適切に統合することで、多品種の製品を効率よく開発できる。

また、対象とする製品系列へSPLEを適用してどの程度の効果が見込めるかの見積もりは、経営判断上必要不可欠である。そのため、SPLEにおける費用対効果 (ROI: Return On

Investment) を推定するためのモデルが考案されている (図-8参照)。さらに、パラメータの不確実性を考慮した推定や、コア資産への再投資タイミングを判断するためのモデルが提案されている⁷⁾。

◆ 既存システムからの移行

SPLE型開発の移行では、既存システム群の要求分析を行い、新たなアーキテクチャを構築する手法が主に提案されている。しかしながら、既存システムの膨大かつ自然言語で記述された仕様書の分析の効率化や、実績のあるソースコードの再利用に対する要求等、多くの課題が残されている。

図-9に、コード・クローン分析を応用した既存システムの共通性・可変性の分析手法⁸⁾を示す。この手法では、まず市場動向・製品計画をもとに製品系列のプ

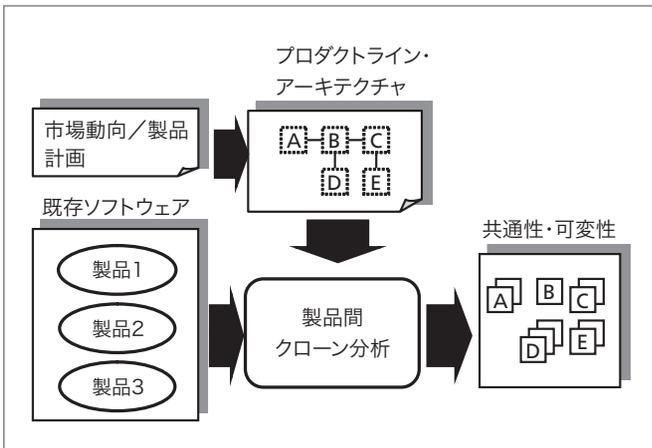


図-9 既存システムの共通性・可変性分析

プロダクトライン・アーキテクチャを設計する。その上で、既存ソフトウェアのソースコードを、プロダクトライン・アーキテクチャ上のコンポーネントにマッピングする。そして各コンポーネントの製品間でのコード・クローン分析を行い、既存製品間での実装の共通性・可変性を定量的に評価する。

今後の展望

ソフトウェアの製品系列を横断した共通化技術として、SPLEを紹介した。組み込みシステムは我が国が競争力を持つ産業の多くを支える重要な技術となっているが、近年の機能増加・市場拡大に伴い、ソフトウェアの大規模化・多品種化が大きな課題となっている。このような状況下で、SPLEは製品系列を横断したソフトウェアの再利用技術として活用が期待される。

SPLEの大きな特徴として、単独の製品ではなく、製品系列を対象として再利用性の向上を図っていることが挙げられる。そのため、技術的な課題はもちろん、ビジネスとしての側面との連携を考慮した研究活動が期待される。すなわち、「製品系列」の範囲をどのように設定するのか、市場・技術予測に基づく製品ロードマップからどのように将来の可変性を推定するのか、市場ニーズの特性から再利用可能資産によるカバー範囲をどのように設定するのかなどのテーマがある。また、現在の研究は主に要求分析および開発工程に焦点があたっているが、今後は検証工程におけるSPLEの応用が必要である。さらに、SPLE運用支援のためのモデルベース開発環境の研究も期待される。

現在、SPLEは組み込みソフトウェアでの事例が多いが、エンタープライズ系での適用事例も少しずつ増え始めている。今後は、エンタープライズ系固有の課題の分析・研究も必要となってくるであろう。

本稿がSPLEの概要と特徴を伝えており、多くの方が関心を持ってくださることを期待する。また、2007年9月10日～14日にはソフトウェア・プロダクトライン国際会議が京都で開催される。今回は初めてのアジア地区での開催である。SPLEに興味を持たれた方の参加を期待する。

謝辞 本解説記事の執筆にあたり、北陸先端科学技術大学院大学 岸 知二先生には多くの有益なアドバイスをいただきました。心から御礼申し上げます。

参考文献

- 1) Linden, F.: Software Product Families in Europe: The Esaps & Café Projects, *IEEE Software*, Vol.19, No.4 (2002).
- 2) Clements, P. and Northrop, L. M.: *Software Product Lines: Practices and Patterns*, Addison-Wesley (2001).
- 3) 進藤智則, 山田剛良, 大槻智洋: プロダクトラインで10倍の格差—ソフト工学にビジネスの視点を, *日経エレクトロニクス*, No.924 (2006).
- 4) Pohl, K., Bockle, G. and Linden, F. V. D.: *Software Product Line Engineering: Foundations, Principles And Techniques*, Springer-Verlag (2005).
- 5) Kang, K. C., Lee, J. and Donohoe, P.: Feature-Oriented Product Line Engineering, *IEEE Software*, Vol.19, No.4 (2002).
- 6) Gomaa, H.: *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*, Addison-Wesley (2004).
- 7) Ganesan, D., Muthig, D. and Yoshimura, K.: Predicting Return-on-Investment for Product Line Generations, *10th International Software Product Line Conference (SPLC 2006)* (2006).
- 8) Yoshimura, K., Ganesan, D. and Muthig, D.: Defining a Strategy to Introduce Software Product Line Using the Existing Embedded Systems, *6th ACM & IEEE Conference on Embedded Software (EMSOFT06)* (2006).

(平成18年12月25日受付)

吉村健太郎 (正会員)
kentaro.yoshimura.jr@hitachi.com

1976年生。1999年早稲田大学理工学部機械工学科卒業。2001年同大学院理工学研究科機械工学専攻修士課程修了。同年(株)日立製作所日立研究所入社。2005年より仏国Hitachi Europe。組み込み制御システムの研究に従事。ACM, IEEE, SAE, 日本機械学会各会員。