

SPLC 2008 概要紹介

山内 和幸

— 組込みシステム開発を現場から支援する —



カンファレンス概要 (1)

■ Software Product Line Conference 2008 (SPLC 2008)

- 概要
 - SPLに関する研究者／実践者が一堂に会する国際会議
 - 企業での経験論文が比較的多いのが特徴
- 期間: 2008/09/08(月)～12(金)
- 場所: Limerick (アイルランド)
- 参加者: 223名 (研究者 120名 + 産業界 103名)
 - 日本からは17名の参加
 - 日立、三菱電機、東芝、リコー、富士ゼロックス、エプソン等から参加
- プログラム詳細: <http://www.lero.ie/splc2008/home.html>

■ 参加者数はほぼ例年並み

- 今年で12回目になるせいか、SPLに精通した人の参加が目立つようになってきた
- とはいえ、導入編等のビギナー向けチュートリアルもこれまでどおり開催されているので、まだまだ新規参加者も多いようである

カンファレンス概要 (2)

■ 今年の経験論文は、“小粒”なものが多かった

- SPLEは企業戦略と結びついた手法なので、企業での適用事例とその効果の発表が、本カンファレンスの醍醐味の1つ
- そういった意味で、「これは！」と唸らせるような、Hall of Fameに繋がる発表が少なかったのは残念であった
- 逆に言えば、欧米では既に普及期に入り、“新技術の採用に積極的な企業”は第一次導入を終えてしまったのかもしれない
 - これは勘繰りすぎか？

■ アジアからは、まだまだ事例発表が少ないのが実情

- 日本、または韓国から研究論文は出ているが、経験論文はあまりない
- その他の国(中国等)に至っては、参加者すらいらない
 - 豊富な人的資源がある内は、開発の効率化には目が向かないか？

カンファレンス概要 (3)

■ 嬉しいニュース

- 昨年ノミネートされた(株)東芝が見事Hall of Fameを受賞！
 - 日本企業初の受賞、おめでとうございます！！

■ 次回はSan Francisco (US)で開催

- 今年よりも経験論文の採用を増やす予定とのこと
 - 論文投稿の〆切：2009/02/20
- SPLEに興味のある方は、是非参加してみたいかがでしょうか？
- 詳細：<http://www.sei.cmu.edu/splc2009/>

印象に残った発表の紹介

- 全プログラムの内、私が特に興味深く感じた5つについて、その概要を紹介します
 - 詳細は、以降のスライドをご覧ください
- その他について、興味のある方はproceedingsをご覧ください
 - IEEEの電子ライブラリ(IEEE Xplore)から閲覧可能です(有料)

タイトル	要約
キーンोटスピーチ: Software Product Lines in Philips Healthcare	蘭Philips社の医療画像機器群へのSPLE適用の歴史を紹介。 その適用方法や苦勞／教訓等を分かりやすく解説した、今回のカンファレンス 中で最も印象的な発表であった。さすがはキーンोट！
パネルディスカッション: Product Line Scoping in Practice	SPLEの特徴の1つである、スコーピングに関するパネルディスカッション。 様々な意見が出たが、スコーピングの多面性や段階的实施に関する示唆等、参 考となる意見も多く、内容の濃いパネルであった。
パネルディスカッション: Agile and PLE	「AgileとPLEは共存するか？」をテーマとした、グループディスカッション。 近年はAgileコミュニティの方も多く参加しているようで、対立ではなく、双方の利 点を活かした適材適所型の運用を提案する意見が多かった。
経験論文: Experiences with Mobile Games Product Line Development at Meantime	携帯電話向けゲームソフトへのSPLE適用事例。 規模は小さいが、不特定多数のデバイスに搭載される、ポータビリティが主たる 変動要因となっているソフトでの事例で、希少な適用事例と言える。
経験論文: HomeAway's Transition to Software Product Line Practice	休暇時のハウスレンタルを提供するWebシステムへのSPLE適用事例。 「60日で適用」というサブタイトルが目引いたが、実際にできるかどうかは別と して、適用の成功に必要な要素を知る上で興味深い内容だった。



Copyright(C) eXmotion Co., Ltd. All rights reserved.

4

キーンोटスピーチ: Software Product Lines in Philips Healthcare

Mr. Luc Koch
(Philips Medical Systems)

— 組込みシステム開発を現場から支援する —



Software Product Lines in Philips Healthcare (1)

■ 概要

- 蘭Philipsの医療画像機器群へのSPLE適用の歴史を紹介

■ Philipsでは既に、以下のPL開発が実現されている

- 10の製品グループ
- 300+の製品
- 1000+の製品バリエーション

■ 規格化 (standardization) とユーザビリティが重要項目

- コーポレートスローガン “sense and simplicity”
- これを実現するために、ユーザインタフェース (UI) の統一は極めて重要

■ 適用方式は集約型アプローチ (centralized approach)

- コア資産開発チームによる、資産ベースの構築
- 一気にPL化するのではなく、step by stepで移行

Software Product Lines in Philips Healthcare (2)

■ PL開発のゴール

- 開発効率の向上
- 製品の類似性の向上 (規格化が狙いであるため)
- 品質と信頼性の向上
- 市場への製品投入間隔の短縮

■ 成功基準

- 再利用資産 (コンポーネント) は、
 - 高品質か?
 - 製品開発に役に立つか?
 - 自由に使えるか?
 - 使いやすいか?
- 加えて、
 - 資産の再利用のためのツールは十分に揃っているか?

■ Philipsでは、上記ゴールに対する成功基準が達成されたため、SPLEの適用が成功したと認知されている

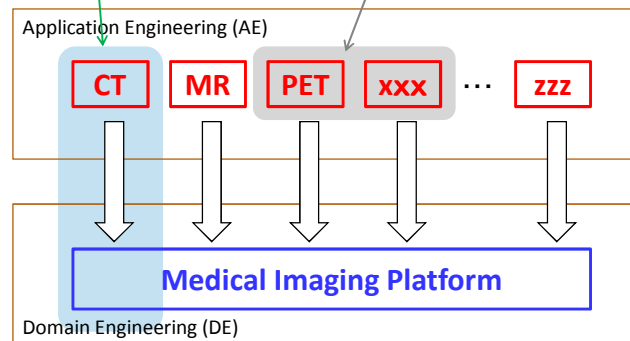
Software Product Lines in Philips Healthcare (3)

■ PhilipsのPL全体像(概要)

- 医療画像機器全体で、プラットフォームを統一
- 異なる製品群でも、同様の操作性を確保できるよう、特にUIのフレームワークを標準化

通常は、1製品群のみでPL化する
→ これだけでも苦労している企業は多い

製品アプリケーション間の再利用も、
PFアーキテクチャにより支援される
→ これらはAEでやっているようだ



Software Product Lines in Philips Healthcare (4)

■ PL architecture (PLA) 構築に払った犠牲/コスト

- 異なる、かつ補完的なバックグラウンドをもつ“ベスト”のアーキテクトの招集
 - 他プロジェクトからの引き抜きに相当するため、他のプロジェクト進捗に影響
- 極めて大きなリードタイム
 - 再利用資産構築に時間を要するため(PLの初期投資)
- 変更に対する自由度(PLAが実用になった後)
 - アーキテクチャの維持のため、システムポリシーに反する変更は原則禁止
- 忍耐と献身
 - SPLEは長期的な取り組みとなるため、特にこれが重要
- 報酬と認知
 - 成功時に、推進者をFellowやVice Presidentへ出世させる等

■ 資産の再利用に関する経済モデル

- 再利用資産に対して、誰が対価を支払うか
- PF-製品開発間でのミスマッチを防ぐため、経済モデルも以下のように進化
 - 第一次: 各部署が使った分だけ払う(製品売上に対する利用コンポーネント数から算出)
 - 第二次: 全部署固定(利用数に関わらず、売上ベースで固定レート)
 - 第三次: 使う部署が払う(必要となる機能をPFチームへ開発依頼も可能に)

Software Product Lines in Philips Healthcare (5)

- 近年は資産管理にInnerSourceアプローチを採用
 - InnerSourceアプローチ：企業内でのオープンソースコミュニティ
 - その企業に属する人なら、誰でも参加可能
 - CollabNet/SVNを利用
 - PFチームが資産のオーナーではあるが、誰でも変更可能
 - PFチームは変更を追跡し、正当性を保証
 - コンセプトは“Let it go!”
 - 本アプローチにより、製品開発チーム側の抵抗はなくなった
 - やらされ感から、「自分も参加している」という意識向上へと変化
 - ただし、管理は大変になったようだ
 - 昨年のHPの発表でも、同様のアプローチを採用していた
 - 本当に上手くいくのかは疑問
- 次のステップ
 - コンポーネント単体の再利用 → 組み合わせの再利用へ
 - すなわち、半完成のアプリケーションとしての再利用という形へ
 - ちなみに、総資産の60%がテスト資産らしい
- PLAも進化
 - Component-based Architecture から、SOA/ROAへ
 - SOA: Service-oriented Architecture
 - ROA: Resource-oriented Architecture

Software Product Lines in Philips Healthcare (6)

- PL推進における教訓
 - PLにより得られる利益を、すべてのステークホルダに明確に説明せよ！
 - 全員が「成果物は何か？」、「プロジェクト計画は？」、etc. について知っていることを確認せよ！
 - 再利用資産の広報を行い、更に本当に伝わっているかまで確認せよ！
 - オーナーシップの欠如を避けよ！
 - アプリケーション/ビジネス両面において、オーナーは誰かを明確にせよ！
 - 衝突は、解決されるまで何度でも交渉せよ！
 - 絶対に野放しにしてはならない！
 - 過去のやり方も支援しつつ、新しいアプローチへ移行せよ！
 - 新しいことを皆がやりたくなるような付加価値を提示せよ！
 - 「管理」が最も明白なリスク項目であることを肝に銘じよ！
 - 経営層の支援を取り付けよ！

パネル #1: Product Line Scoping in Practice

— 組込みシステム開発を現場から支援する —



Product Line Scoping in Practice (1)

■ 概要

- PLスコーピングに関するパネルディスカッション

■ パネラの見解が真っ向から対立

- Scoping vs. Architecture: どちらが重要か？
- Technology vs. Economics: どちらの側面が重要か？

■ 議論内容

- スコーピングは経営問題を扱うタスクであり、体系的な意思決定活動
- 「最も」重要なタスクではないが、重要なタスクの1つ
- スコーピングは2つの側面を持つ
 - 製品ポートフォリオ
 - 資産の再利用
- 「何を再利用可能にすべきか？」 → 技術的側面
 - PLアーキテクチャにより、大きく左右される

Product Line Scoping in Practice (2)

■ 議論内容

- 反復的に行うべきもの
 - 1回目は技術視点で(通常、元になる資産があるので)、2回目以降は経済的な視点で実施するとよい
 - 重量級の作業ではなく、頻繁に繰り返されるべき(ドメインによる)
- 「機能」と「境界」の記述が、明確なスコーピングには必要

■ まとめ

- スコーピングがPLの重要なパートを担うことは、疑いようもない事実である
- 経営的視点(製品ポートフォリオ)に対して、技術面からみた場合の最適解を、経営層に説明できるようにする必要がある
- PLポテンシャル分析も含めて、段階的な移行ができるように、スコーピングも反復的にやるのがベターなようだ

パネル #2: Agile and PLE

— 組込みシステム開発を現場から支援する —

Agile and PLE (1)

■ 概要

- PLEとAgileは共存／相反するのかをテーマにしたパネルディスカッションのはずだったが、同テーマでのアクティブ・ワークショップになった
 - 10名程度のグループに分けて、集中討議
 - 各グループの代表が、討議結果を発表

■ 議論内容(どのグループも同様の結果となった)

- 適用する対象によって、有効性が変わる
 - ドメインの安定性が高ければ、PLEが適切
 - 逆に安定性が低ければ、Agileが適している
- そもそもAgileとは？
 - 人によって解釈はまちまち → XP, Scrum, etc.
- Agile系プラクティスは、PL開発に関わらず有効
 - テスト、要求の優先順位づけ、リファクタリング等

Agile and PLE (2)

■ 議論内容(続き)

- PLEは規律を、Agileは自由を求める？ → 必ずしもそうではない
- Agile「変更を受け入れよ」→ 再利用資産が頻繁に変更しては、PLEの効果が出ない
- ただし、バリエーションを作り出す部分においては、Agileは有効かもしれない
 - AgileはApplication Engineering向き？

■ まとめ

- 一括りにPLE/Agileと議論するのはナンセンス
- 両者の良いところをうまくブレンドし、更により手法ができるなら、それに越したことはない

■ SPLC 2009でAgileに関するworkshopが開催される予定

経験論文 #1:

Experiences with Mobile Games Product Line Development at Meantime

Vander Alves

(Fraunhofer IESE)

Tarcisio Camara

(Meantime Mobile Creations)

— 組込みシステム開発を現場から支援する —



Mobile Games PL Development at Meantime (1)

■ 概要

- ブラジルの携帯電話向けゲームソフトベンダMeantime社でのSPLE適用事例
- 携帯ゲームは無数の端末にダウンロードされるため、その差異をバリエーションとして捉えて、SPLEを適用

■ 携帯ゲームドメインの課題

- 各種デバイスへのポータビリティ
 - デバイスの能力に依存した機能の増減や、ローカライズも含む
- 短いtime-to-market: 4~6カ月
 - 再利用資産構築のような、先行投資をする余裕はない
 - イベントに関連したゲームの開発も多い
 - World Cup時のサッカーゲーム ← リリース時期は絶対にずらせない!



Mobile Games PL Development at Meantime (2)

- 採用したアプローチ: Extractive + Reactive型
 - 再利用資産構築の余裕はないので、Proactive型ではできない
 - 既に相応の既存資産を保有しており、かつSPLEのコンセプトと似た共通化やコンフィグレーション手法を独自に実施していた
 - MG2P (Meantime Game Porting Platform)を軸とした派生開発
 - MG2P
 - Mobile Domain Database
 - ポーティングに関するコンフィグレーション情報を、データベースで一元管理
 - プリプロセッサの定義情報等
 - Meantime Basic Architecture (MBA)
 - 実際のゲームのコードと、リソースファイル(画像や音等)で構成
 - コードには、プリプロセッサ情報が組み込まれており、コンフィグレーション可能
 - Meantime Build System (MBS)
 - 各バリエーションに対応するプロパティファイルに記述されたコンフィグレーション情報を基に、コードとリソースファイルからバリエーションを生成
- コンパイル時バインディングを利用したSPLE適用事例に相当
- MG2P構築後は、ポーティングの生産性が飛躍的に向上

Mobile Games PL Development at Meantime (3)

- (講演者の)結論
 - 小/中規模の企業にとっても、SPLEは有効
 - Time-to-marketが短い/投資余力がないため、Extractive + Reactiveアプローチが良い
 - 大抵、SPLEのことは知らなくても、似たようなことはやっている
- 感想
 - ハードウェアが特定されないソフトウェアでの事例は珍しく、興味深い
 - 通常は、製造業での適用事例が多い(HW/SW両方を自社で開発)
 - 適用ストーリーや技術面では、他の多くの事例と同様ではある
 - 変動要因のほとんどがハードウェア制約によるものなので、Extractiveアプローチがよくフィットした感がある
 - 適用リスクを回避する意味でも、既存資産が多く存在する場合は、Extractiveアプローチを取るのがベターだろう
 - ただし、既存資産の品質にもよるが……

経験論文 #2: HomeAway's Transition to SPL Practice

Charles W. Krueger
(BigLever Software)
Dale Churchett, Ross Buhrdorf
(HomeAway Inc.)

— 組込みシステム開発を現場から支援する —



HomeAway's Transition to SPL Practice (1)

■ 概要

- 2大メジャーSPLツールの中の1つ、BigLever Software, Inc.のGearsの適用事例
 - <http://www.biglever.com/solution/product.html>
- 「あなたも60日でできます」と言っていたが、本当か？

■ HomeAway Inc.という、休暇時のハウステンタルを提供するWebシステムを開発／運営している会社でSPLを適用

■ 会社統合時

- 元々、8つくらいあった同業の会社のM&Aを繰り返して大きくなった会社
- 各社共に異なるシステムを利用していた
- ただし、その基本機能の多くは同様のものだった
- 全システム合わせた規模は、180万LOC

■ 最初のシステム統合

- One-size-fits-allアプローチ
 - システム統合による、現システムへの影響を出したくない
 - サイト毎の独立性(自由な機能追加等)を実現したい
- 全システムのコードを集めて1つのロードモジュール化し、実行時に環境認識して使用するモジュールを決定するように設計



HomeAway's Transition to SPL Practice (2)

■ One-size-fits-allアプローチの問題点

- 不要なコードによる、フットプリントの増大
 - あるサイトで稼働するコードは、他のサイトのコードは不要
- ランタイムの切り替えメカニズムの複雑化
 - 新しい機能が増えて、サイト毎の変動が生じるたびに、新しいメカニズムを組み込む羽目になる
- テスト工数の増加
 - 上記2つの問題に起因

→ もはや、複雑度をコントロールできなくなりつつあった！

■ Gearsを利用したSPLEの適用

- Gearsは、3-Tiered SPL Methodologyという独自の手法をツールに実装している
- 変動点の管理と、既存システムからSPLへの移行を強力にサポートできるツールになっている

HomeAway's Transition to SPL Practice (3)

■ 3-Tiered SPL Methodology

- 最下層から順に、段階的な SPL 環境への移行を狙う方法
 - Step 1:
 - Base Tierと呼ばれる、既存資産の変動点管理と自動コンフィグレーション環境の構築フェーズ
 - 既存資産に存在する変動部の情報収集/整理と、変動点のコンフィグレーション用プロファイルを作成し、自動的に必要な資産を構成できるようにする
 - Step 2:
 - Middle Tierと呼ばれる、コア資産構築フェーズ
 - アーキテクチャのモジュール化と、汎用コンポーネントの作成により、コア資産を軸とした開発ができるようにする
 - Step 3:
 - Top Tierと呼ばれる、製品ポートフォリオの管理フェーズ
 - フィーチャモデルを利用して、機能による製品の組み合わせをコントロールできるようにする
 - この層は、技術面よりビジネス視点が強い

■ HomeAwayの例では、これに加えて step 0 が存在

- 全ステークホルダへの、ビジネスインパクトの説明
- これがすんなり行っているところが、日本の適用事例と大きく異なる
 - 恐らくこれが、60日間で移行ができる大きな理由

HomeAway's Transition to SPL Practice (4)

- 感想(60日でできた理由を検討): **青字は講演者の意見**
 - マネジメントの理解: 各部署への計12回のプレゼンテーション
 - 「60日」というのは、マネジメントのOK判断以降の日数
 - **優秀な人材(特にPLアーキテクト)がいて、その確保がなされている**
 - これがなかなかできない → マネジメントの理解があるからできるのか?
 - 実際、HomeAwayのアーキテクトは他でのSPL適用経験のあるエキスパート
 - 前半30日は、キーパーソンへの特別トレーニングに費やしている
 - もとものの構成が、one-size-fits-all
 - 既存資産が豊富にあり、しかも共通部が圧倒的に多い
 - 29のランタイム・コンフィグレーション・メカニズムが入っていたというが、モジュールの切り替えなので、それほど複雑ではなかったのでは?
 - 組み込みのスバゲッティコードと比較すると、容易と思われる
 - 変動パターンのほとんどが、画面セットの one-of-them?
 - 使用言語とロケーション(サイト)を設定すれば、使用する画面セットでパッケージングされるような作りと思われる
 - 要は、コード規模の割には複雑度の低いシステムだったのでは?
 - とはいえ、ツール自体の機能は、アプリケーションエンジニアリング時に極めて有効と思われる
 - 変動点管理 (variability management)
 - 自動コンフィグレーション (automated production)
- 「どの会社でも60日でできます」というのは無理があるが、条件が整って、やる気になればできるという好例