

製品のシリーズ展開を考えよう

中西 恒夫

ここでは、細かい違いを持つ類似の組み込みシステムを開発する場合に有用な再利用技術として、プロダクト・ライン開発方法論について説明する。この方法論は開発体制や開発環境の地ならしにエネルギーを要するため、手軽に導入できるものではない。しかしその考え方は、日ごろの開発作業で用いている記述の見直しなどに応用できる。

(編集部)

1. 組み込みソフトウェアを いかに再利用するか

さまざまな顧客を相手に製品を供給する場合、顧客によって製品に求める機能や性能、予算が異なるため、しばしば複数の製品モデルを開発する必要に迫られます。また、仕向け地の違い(法律や文化、言語、気候、環境などの違い)によっても、異なる製品モデルを用意する必要があります。流行の変化や技術革新に合わせて、消費者を引きつける新しい製品モデルを絶えず開発し続ける努力も必要です。さらに、パソコンのように画一化されたハードウェア環境を持たない組み込みシステムでは、LSIの調達などの都合でハードウェアが変わり、製品モデルのマイナ・チェンジが行われることがあります。これらの結果、

細かい違いを持つ多様な製品モデルが開発されることになり、ソフトウェアに多数の変種が生じることになります(図1)。

複数の製品モデル用のソフトウェアを同時に、あるいは連続的に開発する場合、それぞれのソフトウェアを一から作っては人手も時間もかかってしまいます。過去に作った資産を再利用したいと考えるのは当然のことでしょう。

● 歴史のある再利用技術、ライブラリ

ライブラリは、ソフトウェアそのものと同じくらいの歴史を持つ再利用技術です。ライブラリとは、将来さまざまなアプリケーションで再利用できそうな共通的なコードを集めて管理する、コード・レベルの再利用技術です。ライブラリによる再利用が最もうまくいっている分野は数値計算ではないかと思います。

数値計算には数学という普遍的な世界観があり、この世界観はプログラミング言語の世界観と近いところがあります。このように、一定の普遍性があり、プログラミング言語とのギャップが少ない分野では、ライブラリのようなコード・レベルの再利用技術でも十分な効果を期待できます。

● システムのモデル化と再利用はトップダウンで

しかし、対象とする分野が抽象的になると、問題をどうモデル化し、プログラミング言語の世界でどう実現するかは百人百様です。問題をどうモデル化したのかはコードだけでは分かりませんし、モデルから逸脱した形ではコードをうまく再利用することはできません。このような背景から、「再利用するアプリケーション領域を定めてモデル化し、ソフトウェアの骨組みを設計し、骨組みに合った再利

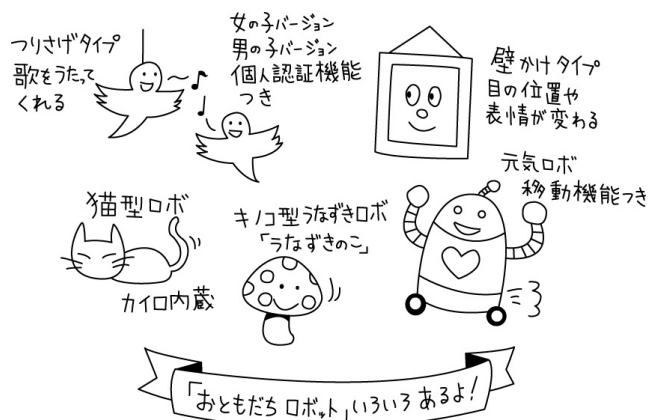


図1 似たような、でも異なるたくさんの製品モデル

用可能なソフトウェア部品を開発する」といった、トップダウン型の大きな粒度の再利用が行われるようになってきました。つまり、ソフトウェア部品のあり方や使い方(=再利用の方法)をソフトウェアの骨組みによって規定し、想定した条件下で確実に再利用できる形にソフトウェア部品を作るのです。

製品モデル間の共通性を活用するソフトウェア再利用技術「プロダクト・ライン開発方法論」も、上述のようなトップダウン型の再利用技術の一つです。プロダクト・ライン開発方法論は米国や欧州では10年以上も前から研究や実践が続けられています。ここでは、プロダクト・ライン開発方法論の基本的な考え方を解説し、その中からとりあえず始められる試みを紹介したいと思います。

2. 5分で分かる プロダクト・ライン開発方法論

一般にプロダクト・ラインといえば「シリーズ化された製品群」を意味しますが、ソフトウェア工学というプロダクト・ラインとは、以下のような特徴を備える製品群を指しています⁽¹⁾。

- (事前の計画で定めた)特定の市場領域に展開する製品群であること
- 共通の特徴や属性(フィーチャという)を持つこと
- ソフトウェア資産(コア資産)を共有し、そこから各製品モデルを構築すること

コア資産を開発することを「ドメイン・エンジニアリング」、コア資産再利用による各製品モデルの開発を「アプリケーション・エンジニアリング」と呼びます(図2)。

● 計画なくして改革なし

アプリケーション・エンジニアリングは、決してその場限りの適当なやり方で開発するものではありません。ドメイン・エンジニアリングの際に、コア資産に対してどのような変形や結合、設定を行って各製品を作り出すのか段取りを決めておき、アプリケーション・エンジニアリングではその段取りに従って製品を開発します。「開発」というより「導出」といった方がよいかもしれません。

このプロダクト・ラインの定義からも分かるように、プロダクト・ライン開発方法論は「今までになかった機能を備えた製品モデルが短期で開発できるようになる、夢のよ

うな方法論」ではありません。むしろ、想定される変化を開発の初期に可能な限り洗い出し、戦略的な再利用のための計画を立て、再利用に適した構造のソフトウェアを開発する方法論だといえます。

● 二つのエンジニアリングの進め方

プロダクト・ライン開発では、ドメイン・エンジニアリングとアプリケーション・エンジニアリングの両プロセスを並行して開発を進めます(図3)。これは開発チームを二つ作らなくてはならないということではなく、同じ開発メンバーが両方のプロセスにかかわっていても構いません。ただし、自分が今どちらの仕事をしているのかを意識しておくことは大切です。

基本的にはまずドメイン・エンジニアリングがあり、アプリケーション・エンジニアリングが進められるわけですが、逆に、アプリケーション・エンジニアリングの際に製品モデル固有の資産を抽出してコア資産化することもあります。

コア資産を増やせばそれだけ保守のコストも発生するので、特定の製品モデルでしか使わない資産はコア資産に含めずに、製品モデル固有の資産とすることもできます。また、ドメイン・エンジニアリングで想定していなかった機能を持つ製品モデルを短期間で開発しなければならない場合や、あるいはよく似た既存製品モデルを集めてプロダクト・ライン開発に移行する(これを「ボトムアップのプロダ

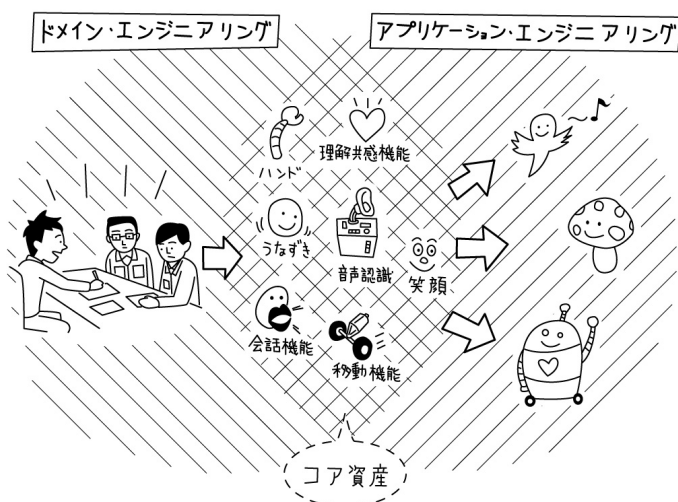


図2 コア資産を中心としたプロダクト・ライン開発

コア資産を開発することを「ドメイン・エンジニアリング」、コア資産再利用による各製品モデルの開発を「アプリケーション・エンジニアリング」と呼ぶ。複数の製品モデルで共有する資産は、すべてコア資産として管理する。