



- **オプションの変異体**：アプリケーションによって、あつたりなかつたりする要素
- **選択肢となる変異体**：複数の選択肢から一つまたはいくつかだけアプリケーションごとに取り入れられる要素

例えば図 1 では、「音声」は対象としている製品系列全体で共通の要素ですが、「音声入力」はオプションなので、その要素を持たないアプリケーションがあることを表わしています。また、「画像」を扱うのは一部のアプリケーションのみですが、その要素があるアプリケーションでは(画像の)「表示」は必須であることも表わしています。そしてその場合、表示するディスプレイのサイズは「1.54 インチ」、「2.5 インチ」、「3.5 インチ」のいずれかから選択することになっています。

## ◆ 「共通」だけれども違うところのある要素の扱い

上に書いたように、共通点であってもその中に可変点を持つ要素があります。純粋に共通な要素であれば、対応する設計要素も共通モジュールなどにしてアプリケーションに組み入れれば良いのですが、可変点はどのように扱ったら良いのでしょうか？ これにはいくつか方法があります。例えば次のようにして設計・実装できます。

- **抽象的な定義を具体的に詳細化する**  
共通に成り立つ抽象的な要素だけを定義しておいて、そこに追加する詳細はアプリケーションごとに改めて定義するやり方。例えば一つクラスを作っておいて、各アプリケーションではそこからサブクラスを作る。また、テストケースで確認する機能を共通に書いておいて、アプリケーションごとにその確認のための具体的な手順を書くなど
- **追加**  
拡張点を用意してそこにオプションで追加要素を組み入れたりする。例えば一連の関数呼び出

しを行なう際に、特定の関数は呼び出したり呼び出さなかつたりするようにする。これは設計時、コンパイル時、実行時など、様々な段階で制御することができる。

- **置き換え**  
インタフェースは同一のまま呼び出される関数の実装を換えることなどで実現される。
- **パラメータ化**  
アプリケーションの振る舞いや特性を決定する要素として数値があった場合、その数値を開発のいずれかの段階で決定することによって、バリエーションを得る。例えばマクロで定義された数値を、ヘッダファイルの選択やコンパイル時の指定によって決定するのはこの方法。また数値でなくても、前述のようにいくつかの関数を用意して、パラメータを介してどの関数が使われるようにするのかを決定するのも、この方法の一種。

## ◆ バリエーションは事業のために

さて、ここで気をつけなければならないことがあります。SPLE は生産性向上の文脈で話題にされることが多いのですが、**真の価値**はその向こうにあります。どういふことかという、**事業の遂行に貢献する**、ということです。いくら多くのアプリケーションを生み出したとしても、それらが役立つ場面がなければ意味がありません。製品ならば、売れるかどうか。業務システムならば、使う部門や会社があるかどうか問題になります。

個々の開発の効率を上げるだけでなく、事業に貢献できるようにするためには技術者チームとマーケティング/企画チームが連携して、自組織の強みが生きるシステム計画を作る必要があります。そして、そのためには技術者が自分達の強みを非技術者に伝える手段を持っていてはなりません。

夢を。



GSLetterNeo Vol. 43

2012年2月20日発行

発行者 ● 株式会社 SRA 産業第1事業部

編集者 ● 土屋正人、柳田雅子、野島勇

バックナンバーを公開しています ● <http://www.sra.co.jp/gslletter>

ご感想・お問い合わせはこちらへお願いします ● [gsneo@sra.co.jp](mailto:gsneo@sra.co.jp)

**株式会社SRA**

〒171-8513 東京都豊島区南池袋2-32-8

夢を。Yawaraka Innovation  
やわらかいのべしゅん