

## 組込みシステムにおける複数の DSL を用いたプロダクトライン開発

徳本 晋<sup>†</sup> 江口 亨<sup>‡</sup> 辻村 浩史<sup>‡</sup> 村上 亮<sup>‡</sup> 伊澤 松太郎<sup>‡</sup> 松本 博郎<sup>‡</sup> 山村 健太郎<sup>‡</sup>(株)富士通研究所<sup>†</sup> (株)富士通コンピュータテクノロジーズ<sup>‡</sup>

## 1. はじめに

近年の市場/ユーザのニーズは多様化・高度化・複雑化しており、企業間競争を優位に進めるには多種類の製品をいち早く提供することが求められる。ソフトウェアプロダクトライン(SPL)はこのような課題に対するアプローチの 1 つである。SPLでは類似性のある製品系列の共通部をコア資産として再利用することで、多様な製品を効率的に開発する。効果的な再利用の手法としては、コンポーネントメタモデルに基づくモデル駆動開発(MDD)を取り入れた研究がなされている[1]。

我々は複数のドメイン特化型言語(DSL)を用いてそれぞれ異なる可変性を実現したモデルを設計することで、より多様な製品を効果的に開発することに取り組んだ。また、設計工程が進むに連れ、使用する DSL のドメイン特化の度合いを段階的に上げることで、高いコード自動生成率を実現することができた。

## 2. 課題

## 2.1. 多種製品にわたる可変性実現の問題

SPLにおいて、可変性の実現手段としてコード自動生成を用いることがある。これはモデル上に可変性を表現できるため、変更や派生に強いシステムを構築できる。

しかし、コード自動生成にDSLを用いる場合は、その特化されたドメインで表現できない可変性についてはコード自動生成で可変性を実現できず、異なる可変性実現手段を選ぶことになる(図 1左側)。このように 1 つのDSLから実現できる可変性というのは限られるため、製品系列として多くの種類を考える場合に問題となる。

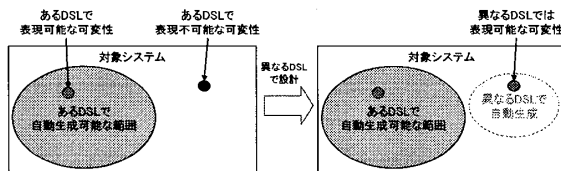


図 1 多種製品にわたる可変性実現の問題とその解決案

Product Line Development using Multiple DSL in Embedded Systems

<sup>†</sup> Susumu Tokumoto, Fujitsu Laboratories LTD.

<sup>‡</sup> Toru Eguchi, Hirofumi Tsujimura, Ryo Murakami, Shotaro Izawa, Hiroo Matsumoto, Kentaro Yamamura, Fujitsu Computer Technologies LTD.

## 2.2. コード自動生成率と汎用性の問題

一般的に汎用性が低い DSL は、そのドメイン特有の情報をメタモデルに多く付加されているためにコード自動生成率が高くなるが、使えるプロジェクトはそのドメインに適合するものに限られる。一方、汎用性の高い DSL は、メタモデルに含まれるドメイン特有の情報が少ないためコード自動生成率が低くなるが、使えるドメインが広がるため多くのプロジェクトで使用可能となる。このようにコード自動生成率とドメイン特化の度合いはトレードオフの関係にあり、双方を満足させることは難しい。

## 3. 提案手法

前章で述べた課題を解決するために、複数の DSL を用いた開発プロセスを提案する。手順は以下になる。

1. SPL を適用する製品系列を定義し、可変性対象について分析する。
2. それぞれの可変性対象に対して可変性を実現できる DSL を検討する。この時、全ての可変性対象について可変性実現可能な DSL を考える必要はなく、既存の汎用性の高い DSL を採用することから考え、設計工程が進むに連れてドメイン特化の度合いが強い DSL を採用し、コード自動生成率を上げていく。
3. (新たな DSL が必要な場合) DSL ツールの開発を行う。
4. 採用した DSL で設計する。その際に残っている可変性対象の可変性実現についても考慮しながら進める。
5. 設計が完了していない場合は 2. に戻る。

設計工程の初期段階は汎用性の高い DSL で大枠を組み立て、ドメインの依存が少ない部分についてコード生成することを目指す。その後設計工程が進むに連れ、ドメイン特化の度合いが強い DSL を採用し、初期段階でコード生成できなかった箇所についてのコード生成をしていく。

この開発プロセスでは、異なるDSLを用いることで多種の製品の可変性を変更に強いモデルで表現でき(図 1右側)、かつ、複数のDSLで相互補完的にコード自動生成することで高いコード自動生成率を実現することが期待できる。

#### 4. 適用事例

今回、提案手法について ET ソフトウェアデザインロボットコンテスト (ET ロボコン) [2] における ライトレースロボットの開発で適用した。

##### 4.1. 2 つの DSL による可変性実現

2009 年の ET ロボコンでは 4 輪型の RCX 走行体と 2 輪型倒立振子の NXT 走行体の 2 種類の走行体を選択でき、我々は 2 チームがそれぞれの走行体でエントリーをした。RCX 走行体と NXT 走行体で基本的なルールは同じで、求められる機能の多くは共通であるため、センサ・アクチュエータなどの各走行体固有の部品を可変性対象とし、それ以外をコア資産として各走行体で再利用して開発を進めた。

また、ET ロボコンではインコースとアウトコースで特徴が異なり、その両方を走行する必要がある。そのため、可変性対象としてその特徴に合わせた走行方法が挙げられた。

この 2 つの可変性対象に対して、RCX 用部品と NXT 用部品の差異という構造に関する可変性を実現する部品化 DSL と、走行方法という振る舞いに関する可変性を実現する戦略 DSL というタイプの異なる 2 つの DSL を使用した。

図 2 に採用した DSL と可変性実現の関係について示す。このように 2 つの DSL を用いることで 4 種類の製品についての可変性を実現したが、例えば異なるセンサ・アクチュエータの製品や異なるコースの製品の種類を増やすことはこれらの DSL で容易に対応可能である。

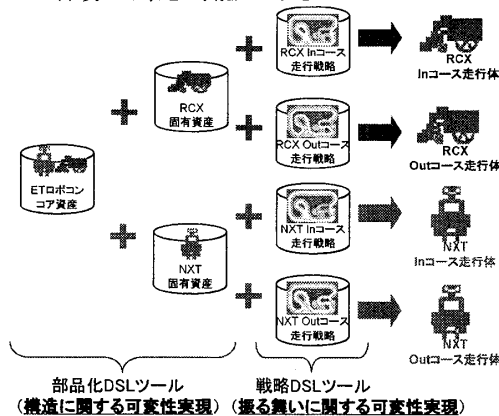


図 2 2 つの DSL による可変性実現

##### 4.2. 部品化 DSL と戦略 DSL

部品化 DSL では各部品についてコンポーネントモデルを設計し、それをインスタンスとして配置した上で、1 周期ごとの各インスタンスの動作順を決めるスケジューリングを記述する。それらのモデルから部品定義コードと全体構造コー

ドを生成できるが、各部品の内部動作についてはハンドコーディングで実装する。

戦略 DSL では走行方法 (アクション) を 1 つの状態と見て、走行方法を変更するときのトリガとなる事象 (ジャッジ) が起こったときに状態遷移するモデルを表現できる。そのモデルからアクションとジャッジの関係を表す戦略データを生成する。

今回は部品化 DSL による設計時に戦略 DSL についての検討を行い、部品化 DSL で戦略データを読み込めるフレームワークを設計した。

各 DSL ツールを用いた設計の流れを図 3 に示す。

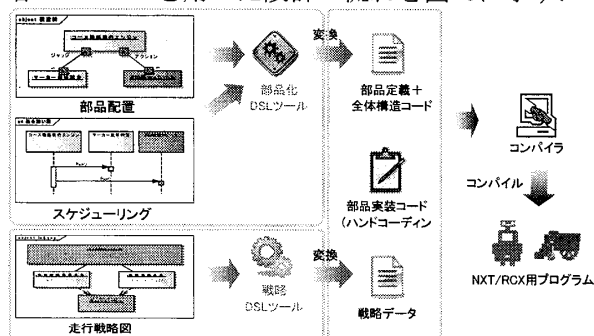


図 3 部品化 DSL と戦略 DSL による設計の流れ

これらの DSL を用いて開発した RCX 走行体のソースコードの全体量とコード自動生成量の比率であるコード自動生成率を求めた結果を表 1 に示す。部品化 DSL のみに比べ戦略 DSL を使うことで 75.6% という高いコード自動生成率が達成できた。また各 DSL によって適切なビューでモデルを記述できたことで、保守性の向上も期待できる。

表 1 RCX 走行体のコード行数と自動生成率

|       | 部品化 DSL 生成分 | 部品化 DSL + 戦略 DSL 生成分 | 全体量       |
|-------|-------------|----------------------|-----------|
| 行数    | 4,567 LOC   | 5,422 LOC            | 7,173 LOC |
| 自動生成率 | 63.7 %      | 75.6 %               | —         |

#### 5. まとめ

本論文では複数の DSL を用いるプロダクトライン開発のプロセスの提案をし、ET ロボコンの適用事例により多種製品にわたる可変性が実現でき、高いコード自動生成率が達成できたことを報告した。今後の課題としては大規模開発に適応できるスケーラビリティの検証が挙げられる。

#### 参考文献

- [1] 加藤滋郎, 後藤祥文, "組込みシステムのプロダクトラインにおける MDA", 組込みシンポジウム 2007 論文集, pp. 54-63, 2007.
- [2] ET ロボコン: <http://www.etrobo.jp>