

組み込みソフトウェアの派生開発におけるコードクローンの安全かつ効率的なリファクタリングパターンの抽出

株式会社富士通コンピュータテクノロジーズ 小林 思無邪 kobayashi.shimu@jp.fujitsu.com

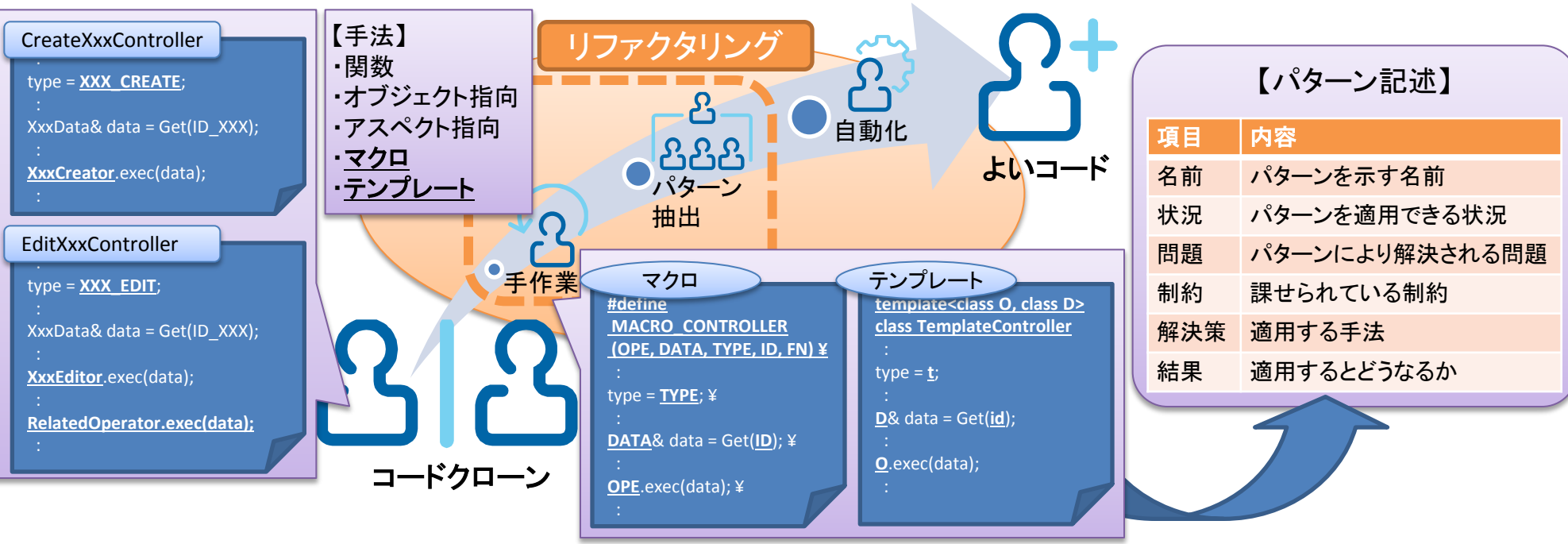
開発における問題点

コードクローンは一般的に保守性を下げるため好ましくないが、組み込みソフトウェアの派生開発においては様々な要因からコードクローンが存在している。しかし、組み込みソフトウェア開発で多く使用されるC/C++言語向けのリファクタリングパターンや支援ツールは少ない。

手法・ツールの適用による解決

現場に導入しやすく、かつ適用できると効率的なリファクタリングが行えるため、C/C++言語特有のマクロやテンプレートを利用したリファクタリングパターンを抽出し、コードクローンを除去するにあたっての指針を明確にする。

リファクタリングパターン抽出のアプローチ



抽出したパターン概要

- マクロ化
 - 型、変数名、関数名が異なるコードクローン(一般にタイプ2と呼ばれる)が存在する状況で、C言語でもメタプログラミングにより解決したい場合に有効
 - 型安全でない、デバッグしづらいなどの問題
- テンプレートクラス化
 - タイプ2のうち型が異なるコードクローンが存在する状況で、テンプレートプログラミングの精通者がいる場合に有効
 - テンプレートに不慣れな場合、保守性が下がる問題
- 暫定対処
 - 短期間での対応が最優先の場合に適用可能
 - 現状のコードクローンを生み出すパターン

まとめと課題

- まとめ
 - C/C++特有であるマクロとテンプレートを用いたリファクタリングパターンを方針レベルで抽出
 - 適用にあたっての特長と問題点を整理
- 課題
 - 方針レベルから具体的なパターンへの詳細化
 - 設計時にパターンの知見を得るために、設計情報からの逆引きができるよう関連付け
 - パターン適用にかかる工数と、適用有無による今後の変更コストの定量化
 - パターン適用の自動化: 適用箇所抽出や判断基準の提示