

分散ネットワーク環境に対する能動的変化メカニズムの適用

2S-5

丸山 勝久 島 健一

{maru,kshima}@slab.ntt.co.jp

NTT ソフトウェア研究所

1 はじめに

従来の部品化再利用において、アプリケーション開発者は、自分のライブラリに存在する部品を自分で変更し、ソフトウェアを開発していた。これに対して、近年の部品組立て型ソフトウェア開発 (Component-Based Software Development) では、開発者 (エンドユーザ) と部品提供者が区別され、開発者はネットワーク上に分散している部品を組み合わせることで、要求を満たすソフトウェアを作成する¹⁾。この場合、部品提供者が、開発者の要求を完全に予測し、必要な部品をあらかじめ準備しておくことは不可能である。このため、部品提供者は、開発者の要求を満たすように機能拡張や機能変更という更新を行う²⁾。しかしながら、分散ネットワーク環境において、開発者各自が現在利用している部品は、開発者のアプリケーション利用環境に応じて、独自の変更が行われている可能性が高い。よって、各開発者が要求する部品は、たとえ類似の機能を持っていたとしてもその実装が微妙に異なり、部品提供者の行う更新だけで開発者の要求に迅速に応じることは困難である。

我々は、開発者の部品変更に対する負担軽減を目的とし、過去の部品変更履歴を用いて他の部品の受けた変更を模倣することで、ライブラリに存在する部品から開発者の要求する機能を満たす新規部品を自動生成する能動的変化メカニズムを提案した^{3,4)}。本論文では、この能動的変化メカニズムをネットワーク上に分散する部品に対して適用する方式を提案し、本メカニズムが分散ネットワークにおける部品変更や部品更新に対しても有効であることを示す。提案する方式は、(a) 開発者の要求を部分的に満たす部品をネットワーク上において配送し、開発者のライブラリに存在する変更履歴と合成する方式、(b) 開発者の要求を満たすために必要な変更履歴をネットワーク上において配送し、開発者のライブラリに存在する部品と合成する方式、の2種類である。

2 能動的変化メカニズム

本論文で扱う部品は、オブジェクト指向プログラムのソースコードである。よって、新規部品の生成とは、開発者の要求を満たす機能を実装したソースコードを生成することを指す。開発者は、ソースコードを直接再利用するのではなく、部品組立て時にソースコードから作成されるオブジェクトコードを組み合わせる。

能動的変化メカニズムは、プログラム・スライシングによる機能分割手法と、プログラム依存グラフの同型写像比較による等価機能の特定手法を組み合わせ、プロ

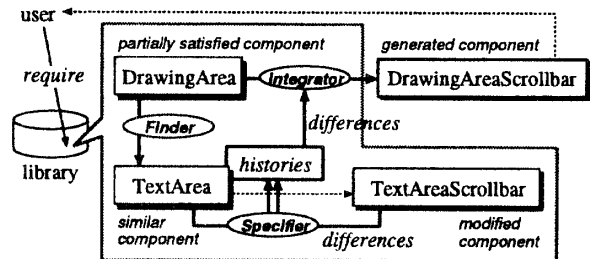


図1: 能動的変化メカニズムによる新規部品の生成

グラム合成アルゴリズムを用いて実現されている。図1に本変化メカニズムによる新規部品の生成の様子を示す。

開発者が、テキスト描画部品 TextArea を変更してスクロールバーを付加した部品 TextAreaScrollbar を作成した場合、変更差異特定手続き (Specifier) は、TextArea と TextAreaScrollbar から変更差異を特定し、変更履歴に蓄積する。次に、開発者が、スクロールバーを備えた図形描画機能を持つ部品を要求した場合、類似部品探索手続き (Finder) は、開発者の要求を部分的に満たす図形描画部品 DrawingArea を見つけ、この部品の持つメソッドとインターフェイスが一致するメソッドを持つ類似部品 TextArea を探す。変更差異合成手続き (Integrator) は、TextArea の変更履歴において、スクロールバーを付加した際の変更差異を DrawingArea に合成することで、開発者の要求を満たす新規部品 DrawingAreaScrollbar を生成する。

このような仕組みを持つ能動的変化メカニズムを分散ネットワーク環境に対して適用する場合、ネットワーク上を配送する対象が部品 (図1の DrawingArea) あるいは変更履歴 (図1の TextArea が所有する histories) によって、2種類の方式が考えられる。2つの適用方式を図2に示す。部品および履歴の両方を配送対象とする場合は、これら2つの方式の組み合わせとみなす。

3 分散ネットワーク環境への適用

本章では、図2に示す2つの適用方式をそれぞれ述べ、実現するための技術について考察する。

3.1 部品を配送する方式

図2(a)に部品を配送する方式を示す。本方式では、開発者の要求を部分的に満たす部品をネットワークを通してリモートライブラリから取得し、開発者のローカルライブラリに存在する類似部品の変更履歴と合成する。一般的に、開発者が行う変更は、アプリケーション利用環境に応じた特殊な修正である。よって、このような変更の履歴は開発者の環境の特性を反映しており、開発者のローカルライブラリに数多く蓄積するべきである。

開発者のアプリケーション利用環境が特殊な場合、例

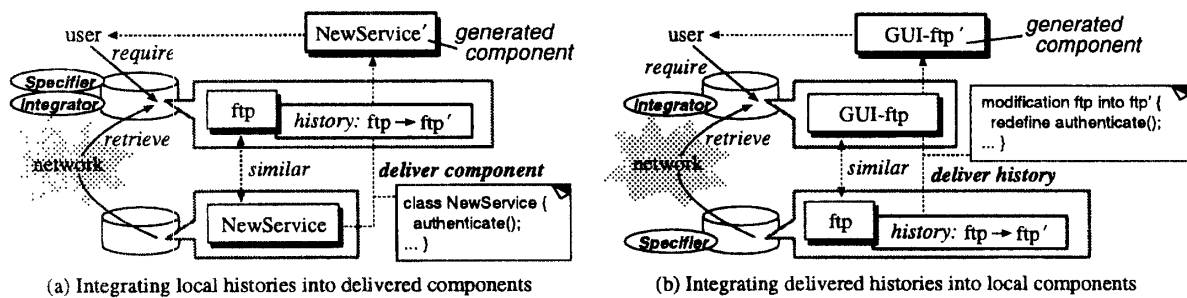


図 2: 分散ネットワーク環境に対する能動的变化メカニズムの適用方式

例えば、開発者の環境ではネットワークアプリケーションに独自の認証機構を採用している場合を考える。この場合、開発者は、汎用部品 ftp などに対して、独自の認証機構を組み込む変更を過去に何度も行っているはずである。いま、新しいサービスに対応した汎用部品 NewService が登場したとき、独自の認証機構を既に組み込んだ部品 NewService' を、この開発者が要求する可能性は高い。本方式では、このような場面において、開発者の環境に特化した変更履歴を汎用的な部品に適用することで、新規部品 NewService' を自動生成する。

3.2 変更履歴を配送する方式

図 2 (b) に変更履歴を配送する方式を示す。本方式では、開発者の要求を満たすために必要な変更の履歴をネットワークを通してリモートライブラリから取得し、開発者のローカルライブラリに存在する部品と合成する。一般的に、部品提供者が行う変更は、さまざまな環境で利用されている部品に対する汎用的な更新である。よって、このような変更の履歴は、開発者の環境に特化した部品に対しても適用できる可能性があり、積極的に公開することに意義がある。

開発者の所有しているアプリケーションが特殊な場合、例えば、独自の GUI を持つ部品 GUI-ftp を利用している場合を考える。いま、従来とは異なる新しい認証機構が提案された場合、部品提供者は、以前に供給した部品 ftp などに対して新しい認証機構を組み込む更新を行い、更新後の部品を供給する。この場合、開発者は、独自の GUI を持つ部品の新しい認証機構対応版 GUI-ftp' を要求する可能性は高いが、このような部品が部品提供者から供給される可能性は低い。本方式では、このような場面において、部品提供者が行った汎用的な変更履歴を、開発者の環境に特化した特殊な部品に適用することで、新規部品 GUI-ftp' を自動生成する。

3.3 考察

プログラムをネットワーク上で検索および配送するという概念は、従来から Java のクラス転送機構などに見られる。よって、方式 (a) は、能動的变化メカニズムを分散ネットワーク環境に対して適用する場合の自然な拡張であり、本メカニズムと従来の部品検索および配送機構を組み合わせることで比較的容易に実現可能である。

これに対して、方式 (b) では、変更履歴をネットワーク上で検索および配送するという新しい概念を含んでい

る。従来も、配送するソースコードの量を減らすという観点で、プログラム全体ではなく更新における差分だけを配送することが行われていた。しかし、変更時の知識を積極的に配送し、各自のライブラリに存在する部品を自動改造するという仕組みは、能動的变化メカニズムを適用することで可能になる。また、方式 (b) を実現する本メカニズムを各ライブラリに組み込むことで、従来部品だけを供給していた部品提供者は、部品変更時の知識の供給という新しいサービスを行うことができる。

我々は、分散ネットワーク環境への適用という観点において、提案した 2 つの方式を実現するために、次に示すような技術が主に必要であると考えている。

- ネットワーク上で変更履歴を検索および配送する技術。特に、変更履歴を検索する場合、IDL (Interface Definition Language) や ODL (Object Description Language) だけでなく、部品の機能や変更時の意味的差分を記述あるいは定義可能な言語が必須である。
- 能動的变化メカニズムが生成するソースコードの単位はオブジェクト指向プログラムのメソッドである。よって、メソッドのようなソースコードの一部分だけを動的にコンパイル可能な技術が必要である。

4 おわりに

本論文では、分散ネットワーク環境に対して、能動的变化メカニズムを適用する際の 2 つの方式を述べた。特に、部品変更時の知識を積極的に供給可能な変更履歴配送方式は、部品提供者の新しいサービスとなる可能性があり、能動的变化メカニズムの有効な応用である。

謝辞 日頃御指導御討論いただく後藤厚宏リーダーはじめ、グループの皆様へ深く感謝します。

参考文献

- 1) 青山幹雄: コンポーネントウェア: 部品組立て型ソフトウェア開発技術, 情報処理, Vol. 37, No. 1, pp. 71-79 (1996).
- 2) Brown, A. W. and Wallnau, K. C.: Engineering of Component-Based Systems, *Component-Based Software Engineering*, IEEE-CS Press, pp. 7-15 (1996).
- 3) 丸山勝久, 島健一: ソースコード再利用における能動的部品変化メカニズム, 情報処理学会論文誌, Vol. 37, No. 12, pp. 2334-2351 (1996).
- 4) 丸山勝久, 島健一: オブジェクト指向プログラミングにおける変更履歴を用いた新規クラスの自動生成, オブジェクト指向'97 シンポジウム, pp. 76-83 (1997).