

LEAD++による動的適応可能なコンポーネントウェアの構成方法

天野 憲樹, 渡部 卓雄

北陸先端科学技術大学院大学 情報科学研究科

概要

近年, インターネットのような解放型分散環境が一般化してきた. また, モバイルコンピューティングのような計算機の利用形態が普及しつつある. 現在, 解放型分散環境およびモバイルコンピューティングで使用されるソフトウェアの開発技術としてコンポーネント技術と移動コード技術が注目されている. これらの技術は併用されることが多くなっており, その例として, JavaBeans にもとづくアプレットがある.

解放型分散環境では多様な実行環境が存在し, モバイルコンピューティングでは計算機のハードウェアやネットワークの構成などが動的に変化する. また, 移動コードプログラムは移動によりその実行環境が動的かつ劇的に変化する可能性がある. こうした変化がプログラムの実行に与える影響は無視できないほど大きい場合がある. しかし, プログラムを作成する時点ではその実行環境や状態変化について正確な情報を完全には得られない.

近年, 多くのソフトウェアはハードウェアやオペレーティングシステムに依存しないミドルウェア上に実装される. このミドルウェアにより実行環境の差異や状態変化はある程度隠蔽される. しかし, その物理的な差異や変化の度合いが大きい場合, その影響は上位のプログラムにもおよぶ. また, ミドルウェアによる実行環境の隠蔽は, 上位のプログラムに各計算機固有の機能(特殊なハードウェア装置など)を利用不可にする, という別の問題を引き起こす.

こうした問題に対し, 各ソフトウェアの振舞いを実行環境の状態に応じて変える手法は有効である. しかし, 各ソフトウェア内部の振舞いを外部から汎用的に制御するのは難しい. このため, 各ソフトウェアが実行環境の状態変化を認識し, それに応じて自ら振舞いを動的に変更できる能力(機構)を持つことが望まれる. このような実行環境に対するソフトウェアの適応能力を動的適応可能性と呼ぶ. 我々は

これまでに動的適応可能性を持つソフトウェアモデル *DAS* とその記述言語 *LEAD++* を提案した.

今回, 我々はコンポーネントウェアに *DAS* モデルを適用し, *LEAD++* による動的適応可能なコンポーネントウェアの実現を試みた. *LEAD++* は独立したひとつのプログラム(コンポーネント)内における適応動作の記述を意図して設計された. つまり, 独立した複数のコンポーネントから構成されるコンポーネントウェア全体に関する適応動作の記述を言語的に支援していない. しかし, コンポーネント間の接続を動的につなぎ換えるスクリプト言語として *LEAD++* を用いることもできる. 具体的には, 1) コンポーネント間の構成(組み合わせ), 2) 各コンポーネント内部の振舞い, をそれぞれ実行環境の状態に応じて動的に変更することで動的適応可能なコンポーネントウェアを実現する.

本稿の構成は以下のとおりである. 2 節では, 動的適応可能なコンポーネントウェアの基本概念を示す. 3 節では, *DAS* モデルおよび *LEAD++* について, その概要を示す. 4 節では, *LEAD++* を用いた動的適応可能なコンポーネントウェアの構成方法について詳説する. 5 節では, 関連研究について簡潔に述べ, 6 節でまとめと今後の予定を示す.