

審査の結果の要旨

氏名 松崎 和賢

アンビエント環境とはユビキタス環境の次の世代の環境として位置づけられ、あらゆる電子的媒体が設置される環境に応じて適応されることを想定している。このためアンビエント環境で動作するアプリケーションの設置（ディプロイメント）は従来の環境におけるディプロイメントに比べて困難になる。これは、アプリケーションがその本来の動作（アプリケーションロジック）を実行しながら環境の変化に対応・環境の資源を利用するために適応動作を取る必要があり、この適応動作がディプロイメントの各フェーズを複雑にすることに起因する。既存手法では適応動作の設計支援が乏しいため実現のコストがかかり、一つの環境に特化した記述になり、さらにアプリケーションロジックと複雑に絡みあい保守性が低下する。

本研究ではこれらの経験則からディプロイメントを困難にする3つの依存性として、空間依存性、局所依存性、コンテキスト依存性の3つを発見している。そして、ディプロイメントの容易性を向上させるため、これらの依存性に対処するSpatio-Temporal Scope (STS)アプリケーションフレームワークを提案、実装し、評価している。

フレームワークはSTS適応動作と動的タスク配置の二つの機能に基づいている。前者は適応動作のモジュール化、後者はアプリケーションを動的に複数ノードへ配置する際に役立つ。STSの名前に含まれる“スコープ”は、適応動作のモジュール単位であり、プログラムの静的・動的な情報と周辺状況とを離散値に抽象化し、適応動作の条件として利用する。これは、適応動作の必要な条件がプログラム内・アンビエント環境内に横断的に現れることに対処し易くする。

記述系は、ノードの位置と提供可能なサービスの情報を管理するサーバと連動し、開発者がノードの配置を抽象的に記述することを支援する。開発時にディプロイメントの対象となる具体的な情報が無い場合での開発を支援する。双方の機能を利用するための入力情報は抽象的・宣言的な時空間モデルを用い、これらの情報は実行時に実体化される。

これは二つの意味を持つ。一つは依存性の要素が実行時に注入されること、つまり依存性がコードから消えディプロイメントに悪影響を与えなくなることである。もう一つは適応動作実現時に必要な外界との相互作用などが動的に合成されることであり、これはプログラミングの負担を軽減することになる。双方ともアプリケーションの更新作業を従来に比べ容易にする。フレームワークの評価に関して、アプリケーション事例に対するSTSフレームワークのコーディング負担への影響を測定し、ディプロイメントの各フェーズを支援することを示している。

以上のように、本論文では、アンビエント環境におけるアプリケーションを適応させるために問題となる依存性について検討を加え体系化し、それら問題点を解決するための枠組みを与えることにより、当該研究分野に多大な貢献を与えた。

よって本論文は博士（情報理工学）の学位請求論文として合格と認められる。