

論文審査の結果の要旨

氏名 ニーストロム ペルソン ヨーハン トマス

本論文は、進化可能なコンポーネント統合を可能とする **Java** の拡張を提案している。拡張は **Poplar** と呼ばれる。

本論文は7章から成り立っている。

第1章では、本論文の背景、本研究に至った動機、そして、本論文の貢献が述べられている。

第2章では、**Poplar** のデザインについて詳しく述べられている。**Java** におけるコンポーネント・ベースのプログラミングやリファクタリングについて述べられた後、**Java** のコンポーネントの進化が分析されている。そして、ラベル付き変数に対するクエリをもとに、**AI** プラニングの技術を応用して、コンポーネントを統合する仕組みが導入されている。この仕組みにより、コンポーネントの進化の後にも、正しくコンポーネントを統合できることが示されている。

第3章では、**Poplar** の言語仕様が、非形式的ではあるが、詳細に述べられている。この章を読むことにより、実際に **Poplar** を利用することが可能になると考えられる。次の第4章では、第3章の言語仕様が **Middleweight Java** の拡張として形式化され、特に、型付けの規則が厳密に与えられている。

第5章では、**Poplar** コンパイラのデザインと実装について述べられている。**Jardine** と呼ばれる処理系は、クエリに対して **AI** プラニングを適用し、コンポーネントを統合することによって、**Poplar** のコードを純粋な **Java** コードに変換する。

第6章では **Poplar** の評価が与えられている。まず、**JFreeChart** と呼ばれるコンポーネント・ライブラリを用いた比較的巨大的なアプリケーションがケーススタディとして述べられている。**JFreeChart** コンポーネントの統合はクエリによって行われ、**JFreeChart** をリファクタリングした後も、コンポーネントの統合が正しく行われることが示されている。次に、**Fowler** のテキストに掲載されているリファクタリングの分類に基づき、各種のリファクタリングに対して、**Poplar** のアプローチが、有効かどうかを検証され、この結果に基づき、**Poplar** のアプローチの評価が与えられている。すなわち、**Poplar** は広範な種類のリファクタリングに対応可能であり、**Poplar** のもとでは、コンポーネントの進化を極めて容易に行うことができる。

第7章では、関連研究との比較とともに、本研究の総括が行われている。

結論として、本論文はソフトウェア・コンポーネントに対する新たなアプローチを提案し、その応用可能性をケーススタディによって示している。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。