The University of Osaka
Institutional Knowledge Archive

Title	レガシーシステムマイグレーション支援のためのソフトウェア分析・設計手法の研究
Author(s)	岡田, 譲二
Citation	大阪大学, 2022, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/88136
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

論 文 内 容 の 要 旨

氏 名 （ 岡田譲二 ）	
論文題名	レガシーシステムマイグレーション支援のためのソフトウェア分析・設計手法の研究

論文内容の要旨

企業においては，ビジネス上は重要だが保守の継続が困難な巨大なメインフレーム上の基幹システムが多数存在しており，その変更には多くの期間とコストが必要になる．この問題を解決するために，異なるプログラミング言語に機械的に変換するマイグレーションがしばしば行われる．システムをマイグレーションする戦略としては Chicken Little 戦略がよく知られている．
本研究では，著者が所属する企業において実施したマイグレーションプロジェクトにおいて実際に直面した課題とその解決策について述べる．このマイグレーションプロジェクトにおいては，Chicken Little 戦略におけるレガシーシステムの分析とレガシーシステムの構造の分解，新アプリケーションの設計について，課題が発生していた．
レガシーシステムの分析では，分析の前提として対象のソースコードファイルのプログラミング言語が判明している必要があるが，実プログラミング言語が不明なソースコードファイルが多数存在しており，この判定を手作業で行うと多大な労力が必要となる．本研究では手作業で判定すべきファイルの代表をクラスタリングによって選出する手法を提案し，他手法と比較評価した．その結果，99.49%のファイルを正しく分類できることを確認した．
レガシーシステムの構造の分解では，業務用語とシステムの用語の対応関係を把握することが重要である．本研究では，ソースコードを処理機能という粒度で分割し，入出力されるファイル名と処理機能から機能名を自動的に付与する手法を提案した．定性的な評価を行い，本手法によって業務用語とシステムの用語の対応関係が把握しやすくなったことを示した．
新アプリケーションの設計では，マイグレーション後の性能問題という課題があった．本研究では，新しいプログラミング言語に書き換えられたプログラムを並列実行可能な形に書き換える作業を支援することで，移行後のプログラムの実行時性能を改善する手法を提案した．評価実験として，書き換え前後のプログラムの実行時間の評価を行った．本手法で，書き換え前と比較して実行時性能を 2 倍から 50 倍改善させることができた．
これらの手法により，工数や予算が大きくかかりがちな企業におけるレガシーシステムのマイグレーションプロジェクトにおいて，一定の効率化を果たすことができたと考えられる．

様式 7

論文審査の結果の要旨及び担当者

氏　名　（岡田譲二）			
論文審査担当者		（職）	氏　名
	主 査	教授	井上 克郎
	副 査	教授	楠本 真二
	副 査	教授	八木 康史

論文審査の結果の要旨

本学位論文を審査担当者間で精査した結果、以下に述べる内容を確認した。

本学位論文は、レガシーシステムマイグレーション支援のためのソフトウェア分析・設計手法に関して３つのステップに対して以下の３つの研究を行った。

1つ目は、学位論文の2章で示すように、レガシーシステム中のソースコードファイルのプログラミング言語を判定する作業を支援する手法を研究した。本手法ではパターンマッチとクラスタリング、静的解析の結果を組み合わせることによってプログラミング言語の判定を支援する手法を提案した。複数の結果を組み合わせることで高速に精度よくプログラミング言語を判定することが可能となる。提案手法は人手による判定よりも大幅な工数削減を行うことができ、既存手法と比べて高い精度で判定できた。この手法を用いることで、分析者はレガシーシステムのプログラミング言語の判定を小さい工数で実施することが可能となることが示された。

2つ目は、3章で示すように、変更要件に関係するプログラムを特定するために業務用語と対応関係が取りやすい処理名を付与する手法を研究した。本手法では、ソースコードを処理機能という粒度で分割し、入出力されるファイル名と処理機能から機能名を自動的に付与する手法を提案した。定性的な評価により、本手法によって業務用語と処理名の対応関係の把握が容易になることが示された。

3つ目は、4章で示すように、マイグレーションに後のプログラムを並列実行可能な形に書き換える作業を支援することで，マイグレーション後のプログラムの実行時性能を改善する手法を研究した。ループを用いたデータの処理をパターンとして抽出し、抽出結果とバッチフレームワークの各構成要素との対応関係を明らかにした。この関係を基にバッチフレームワークを利用するようにリファクタリングすることで、並列処理を行うことができるようになり、実行時性能を改善することが可能となる。本手法によるリファクタリング前後のプログラムの実行時間とリファクタリング工数の評価を行い、実行時性能が改善させることとリファクタリングの工数が削減されることが示された。

本論文で得られた知見は、レガシーシステムを移行する実務者や、レガシーシステムの移行支援手法についての研究者に非常に役立つものである。よって、博士（情報科学）の学位論文として価値のあるものと認める。