チケット利用プロセスに着目した チケッティングシステムのソフト可変性分析

小林悠一[†1]　望月智之[†1]　志村明俊[†1]　吉村健太郎[†1]　茂木栄[†1]

社会インフラシステムは長期運用の間，ユーザニーズに応えるため繰り返し改修が行われてきた．チケットの販売・利用やゲートの入退場状況等を管理するチケッティングシステムでは，初期導入以降様々な高機能化拡張を行ってきたことにより，ソフトウェア構造の複雑化を招いている．チケッティングシステムの持続的な維持保守のため，ソフト構造の見える化と開発手順の標準化が必要となる．このような課題に対し，チケット利用プロセスに着目したソフト可変性分析手法を提案する．本方式では，チケット利用のプロセスをチケット情報授受とサービス利用判断から成ると仮定，サービスを実現するプログラムとチケット利用プロセスの軸でソフト構造を見える化した．次に，基本機能のプログラムについて共通部と可変部を見出し，それぞれテンプレートとコード自動生成を用いた開発手法を作成した．標準開発手順に本方式を用いた場合，対象機能の約７割のコードが自動生成可能である見通しを得た．今後は，その他インフラシステム等への本手法の応用・展開に向けた研究を進めてゆく．

1. はじめに

チケットの販売・利用やゲートの入退場状況等を管理するチケッティングシステムは，チケット媒体の技術変化に合わせて機能拡充を繰り返してきたため，ソフト構造が複雑化している．継続的な維持保守のためには，段階的なソースコードの置き換え(クレンジング)をソースコードの可変性を見極めつつ効率よく進める必要がある．

本論文では，チケット利用プロセスに着目したチケッティングシステムのソフト可変性分析を提案する．チケット業務プロセスモデルの可変性を着眼点としたソフトウェア構造の可変性分析を行い，チケッティングシステムの可変性モデルを得た．

本手法を適用した場合のクレンジング作業の開発効率を評価するため，可変性モデルに基づくソースコード自動生成ツールを作成し，自動生成率を評価した．結果，コード置き換え時の自動生成率約７割を得た．

今後は他システムへの本手法の応用・展開に向けた研究を進めてゆく．

2. チケッティングシステムの課題

2.1 概要

チケッティングシステムは，チケットの新規発行，利用者とのひも付けからチケットの登録状態や利用状況を保持・管理し，最終的には利用者情報抹消した上でチケットの廃棄を行うまでの一連のライフサイクルを長期に渡って管理するシステムである．

このチケッティングシステムは，オンライン系とバッチ系の機能を基本機能として構成されている（図 1）．オンライン機能は一般利用者やオペレータといったユーザの要求に応じデータ登録・参照を行うものであり，例えば券売機によるチケットの新規発行やオペレータ端末からのチケット利用者情報参照といった機能を実現する．一方，バッチ機能ではチケット利用情報等の整合性を確認し，チケットに関する情報の維持・利用状況の整理を行っている．

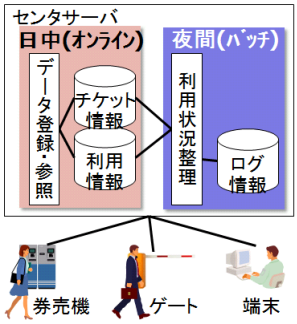

図 1　チケッティングシステム構成

2.2 課題

チケッティングシステムが扱うチケットは，磁気ストライプやバーコードを利用したものから，近年では携帯電話内臓のＩＣチップや，指静脈等生体認証を用いたものまで時代を経て変化している．このような変化に対応するために繰り返し改修を行った結果，チケッティングシステムのソフトウェアでは構造複雑化が起こっている．

継続的な維持保守のためには，段階的なソースコードのクレンジングをソースコードの可変性を見極めつつ効率よく進める必要がある．

2.3 従来技術

ソフトウェアの構造整理や持続的な維持保守に関する従来研究としては，リファクタリングや SPL(Software Product Line) [1]がある．SPL ではソフトウェアの部品を未来の製品展開を見据えて設計するというものであり，エンタープライズ系システムでも非機能要求（性能・拡張性）に対する担保を考慮した SPL 開発手法[2]等が提案されている．また，SPL は基本的にはスクラッチ開発を前提としたものであるが，旧プログラムの機能観点での分析とアスペクト志

†1 (株)日立製作所
Hitachi Ltd.

向での分離による SPL 化のアプローチ[3]も提案されている．しかし，これらの方法は新規開発やスクラップ＆ビルドを前提としており，段階的な置き換えに利用した場合には，効率的な実現は難しい．

3. チケッティングシステムの可変性分析

3.1 R^3 プロセス応用による可変性分析

長期間運用を前提とするシステムの段階的刷新を実現するためのフレームワークとして，本研究ではこれまで R^3 プロセスを提案してきている[4]．R^3 プロセスは Reform, Refine, Renovation の 3 ステップから成り，このうち Reform, Refine ステップを応用することにより，段階的コード置き換えが可能になる．

図 2 (左)に Reform, Refine ステップの概要を示す. Reform ステップは仕様・実装分析として仕様書やソースコードといった既存の資産をトレーサビリティマトリクス，フィーチャマトリクス，XDDP[5]等の手法を用いて分析・見える化し，システムや機能間で比較を行った結果として対象システムの特徴を抽出する. Refine ステップでは Reform ステップで抽出したシステムの特徴から対象システムのモデルを作成，モデルから自動生成方式を確立する．このステップまで情報制御系のソフトウェアに実施した例として RASYS[6]が挙げられる．なお，Renovation ステップでは最終目標として Reform，Refine ステップまでで得た知識からシステムの抜本的刷新を行う．

本論文では，クレンジングを目的としたチケッティングシステムへの Reform，Refine ステップ応用，及びチケット業務プロセスモデルを加味した可変性分析について提案する（図 2）．

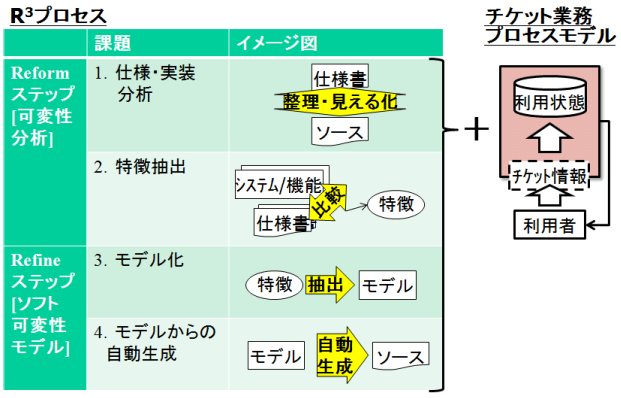

図 2　Reform,Refine ステップとチケッティングシステム適用観点

3.2 チケット業務プロセスモデル

チケット業務プロセスは，チケット情報授取とチケット利用状態確認の一連の流れを持って実施されている．チケット情報授受は利用者からチケットの情報を取得する部分であり，サービス変化の影響を受けない．一方，チケット利用状態確認は，サービスの拡充・変更毎に改修が必要である．

本論文では，業務の可変性の特徴がソフトウェアの可変性の特徴と密接に結びついているのではないかと仮説を置き，これをチケッティングシステム可変性分析の着眼点として，以降の分析を行う（図 3）．

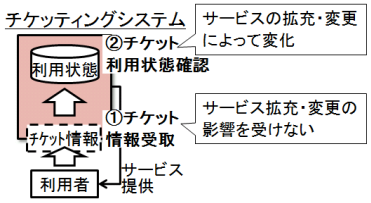

図 3　チケット業務プロセスモデル

3.3 チケッティングシステムの可変性分析

現行の構成見える化のため，仕様書とプログラムとの整合性確認と機能間の関連性・類似性整理の手法であるトレーサビリティマトリクスを用いた可変性分析を行った．

結果を図 4 に示す．各機能の構成要素となる関数は機能間で互いに類似，または同名の関数を重複保持していた．これらの関数を業務プロセスモデルのチケット情報受取，チケット利用状態確認のプロセスに対応づけると，全ての関数は 2 つのプロセスのいずれかに分類可能であった．さらに，チケット情報受取には電文チェックと返信電文作成，チケット利用状態確認には DB 検索・更新とチケット状態検定の関数群が分類された．

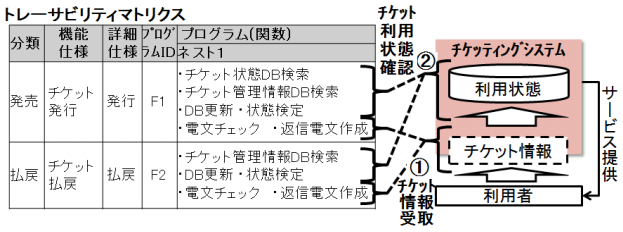

図 4　トレーサビリティマトリクスを用いた見える化

上記 4 種類の関数群における，機能間の共通性についてフィーチャマトリクスを用いた可変性分析を行った(図 5)．チケッティングシステムの全機能を対象に分析を行った結果，チケット情報受取に対応する電文チェックと応答電文作成はほぼ必ず存在，チケット利用状態確認に対応する状態検定と DB 検索・更新の関数は多くの機能で利用されていることがわかった．そこで，これら 4 種類の関数をチケッティングシステムに共通な関数群として抽出した．さらに，これらに対し可変性を整理・モデル化すれば，4 種の分類毎での効率的クレンジングのためのポイントが抽出できる．

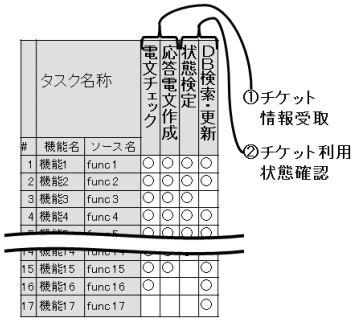

図 5　フィーチャマトリクスによる共通性確認

3.4 ソフトウェア可変性モデル

図 6 にチケッティングシステムソフトウェア可変性モデルを示す．チケッティングシステムは，サービス毎に不変なチケット情報受取と，サービス毎に可変なチケット利用状態確認に分けられる．以下，それぞれに対応した関数群に対し，構造の特徴をモデル化する．

電文チェックと応答電文作成の２種類の関数群の構造については，対応するチケット情報受取がサービスの拡充・変更に影響を受けない部分であることと，実コードの構成比較により，バリエーションが不変の定型チェックを必要に応じて選択可能なパタン選択方式としてモデル化した．

DB 検索・更新と状態検定ではチケット利用状態確認に対応し，サービス変更・拡充による改修が起きることからサービス毎可変となる．この２種類の関数群はそれぞれサービスに応じた DB アクセスと，DB 情報を用いたサービス利用状態判断を行うことから，DB 検索・更新は DB アクセス部分のみ変更可能な SQL 生成表＋テンプレート方式，状態検定は DB アクセス結果に対する判断を中心に行う決定表方式としてモデル化した．

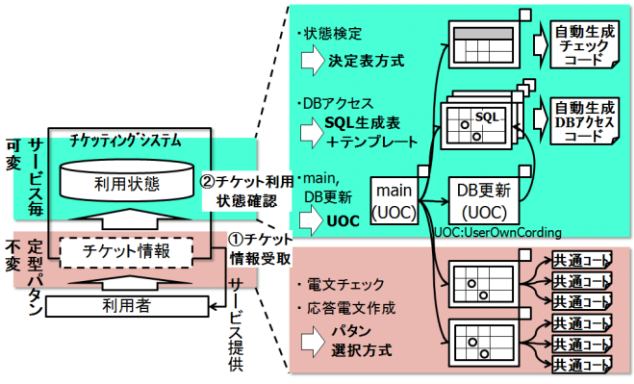

図 6　チケッティングシステムソフト可変性モデル

4. 評価

4.1 適用方法

本手法を適用した場合のクレンジング作業の開発効率を評価するため，可変性モデルに基づくソースコード自動生成ツールを作成し，自動生成率を評価した．

4.2 自動生成ツール

チケッティングシステム可変性モデルを用い，自動生成を行う．自動生成の基本方式として，詳細設計レベルで記される処理定義を表形式のフォームとして記載, 表を XML に変換，さらに変換プログラムからソースコード生成を行う（図 7）．この方式をチケッティングシステムの４種類の関数に対応した３つの方式に合わせて適用し,ソースコード自動生成を実現する．

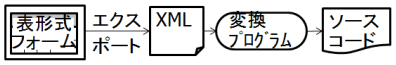

図 7　自動生成の基本形

4.2.1 パタン選択方式[電文チェック，応答電文作成に対応]

パタン選択方式では機能仕様書に記載されている詳細レベルのチェック項目に対応したコードブロック呼出しをパラメタとして選択する（図 8）．

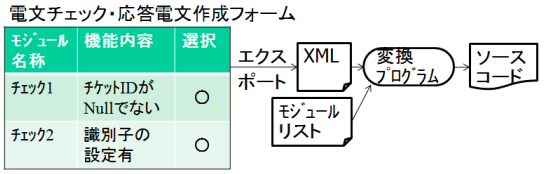

図 8　パタン選択方式

4.2.2 決定表方式[状態検定に対応]

決定表方式では if 文や switch-case 文といった判定文を決定表で作成し，チェックロジックの定義を行う（図 9）．

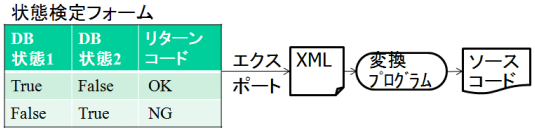

図 9　決定表方式

4.2.3 SQL 生成表＋テンプレート方式[DB アクセスに対応]

SQL 生成表＋テンプレート方式では，変更がない部分はテンプレート化しているため，変更部分をフォームを用いて定義し，差し替える形で自動生成を行う．（図 10）．

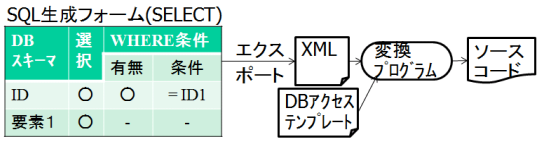

図 10　SQL 生成表＋テンプレート方式

4.3 評価試験

以下を対象とした評価試験を行った.

・対象機能　：チケッティングシステムのオンライン機能
　　全 43 機能中 3 機能
・ステップ数：3 機能合計約 3.5k ステップ

4.4 結果と考察

対象 3 機能の実行ステップ数に対し，本方式を適用したところ,各関数を呼び出す main 関数や変更部分局所化構成を採用した DB 更新の一部でフォームからの自動生成ではカバーしきれない部分があった．この部分については予め，開発者が手を入れる前提の UOC（User Oriented Code）として切り出した．その結果,UOC 以外の自動生成率約 7 割を実現した．この方式は，前述のようにチケッティングシステムのモデルに基づいたものであるため，他機能に対しても同等の効果が期待できる．

また，これらのフォームを利用した自動生成方式を開発プロセスに応用する場合には，このフォーム自体を仕様書に組み込むことで，機能設計からそのままコード生成を行うことになり，設計と実装の工数を大幅に低減できる（図 11）．また，コーディングの後工程に関しては，自動生成部のテスト工数の低減, 不具合発生時の原因箇所限定容易化，パタン化したコードによる可読性向上といった効果が期待できる．

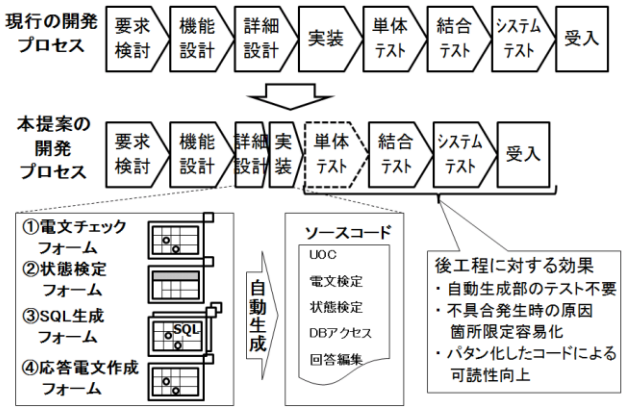

図 11　自動生成による開発手順の変化

5. まとめ

チケット業務プロセスモデルを着眼点に用いた可変性分析をチケッティングシステムに対して実施し，クレンジング対象となるソフトウェアの単位を抽出した．抽出結果に対し可変性のモデル化と自動生成による低工数でのクレンジング方法をツール化，評価試験を行いソースコード自動生成率約 7 割という結果を得た．

今後は，R^3 プロセスの最終ステップである，Renovation ステップ実現に向けた深度化，及び他分野への本手法の展開に向け研究を進めてゆく．

参考文献

1) クラウス・ポール他：ソフトウェアプロダクトラインエンジニアリング ソフトウェア製品系列開発の基礎と概念から技法まで, エスアイビーアクセス，2009
2) 石田：エンタープライズシステムにおけるソフトウェアプロダクトラインの適用, 情報処理 Vol.50 No.4 Apr. 2009 .
3) Kwanwoo Lee 他：Combining Feature-Oriented Analysis and Aspect-Oriented Programming for Product Line Asset Development, 10th International Software Conference, 2006
4) 飯島他: 社会インフラを支えるシステム技術，日立評論, Vol.93, No. 12, pp.832-837，2011 年 12 月
5) 清水：「派生開発」を成功させるプロセス，技術評論社, 2007
6) 大沼他：情報制御システム記述言語 RASYS によるモデル記述の網羅性検査ツールの開発，研究報告ソフトウェア工学（SE） 2011-SE-171(9), 1-8, 2011-03-07