

モデルベーステストに基づく結合テスト全自動化の試み

丹野 治 門^{†1} 張 暁 晶^{†1} 星野 隆^{†1}

モデルベーステストの考え方にに基づき、ソフトウェア設計書からテストで必要となるテスト項目、テストデータ、テストスクリプトを網羅的に自動生成することにより、業務システムの結合テスト全自動化を実現するアプローチについて議論する。

Toward Full Automatic Integration Testing via Model Based Testing

HARUTO TANNO,^{†1} XIAOJING ZHANG^{†1} and TAKASHI HOSHINO^{†1}

The authors would like to discuss on an approach to exhaustively generate test case, test data and test script based on model-based testing technique toward full automatic integration testing for enterprise systems.

1. モデルベーステストに基づくテスト全自動化

テスト工程は全体のコストのうち 50% を占めるとも言われているが、現状、その多くが人手で行われており、コスト削減の余地があるといえる。本研究ではこれまで手動で行っていた作業を機械で置き換え、図 1 に示すような、テスト設計、テスト実施が自動化された世界を目指す。

提案するアプローチ [2] はモデルベーステスト (以降、MBT と呼ぶ) の考え方にに基づいており、テスト工程の前工程である設計工程の成果物であるソフトウェア設計書をモデル化して入力としている。そして、ソフトウェア設計書から網羅的なテスト項目及びテスト手順、テスト用データ、テストスクリプト等を自動で生成する。また、テスト設計において生成したテスト用データ、テストスクリプトを利用し、既存のテスト実施ツールと連携させることで最終的には「ボタン 1 つでテスト設計からテスト実施までが全て自動で行われる世界」を目指している。本稿では、業務アプリケーションのテスト支援にスコープを絞る。そして、テスト項目の数が多く、自動化の効果が見込まれる結合テストに着目し、特に業務アプリケーションの結合テストに関するテスト設計自動化の取り組みについて説明する。

2. スコープとする業務アプリケーションの結合テスト

業務アプリケーションの結合テストとは、図 2 に示すように、Web ブラウザ、サーバ側の処理、データベースを結合させたときに、システムが正しく動作するかを確認するためのテストである。例えば、図の例だと「検索画面」で検索を行い、正常に検索が行われ検索結果画面へ遷移する動作や、エラー画面へ遷移する動作などがあるが、テスト設計においてはこれらの動作のバリエーションを漏れなくテスト項目として抽出する必要があり、ログイン ID などの入力バリエーションに関するテストのバリエーションなども必要となる。それぞれのテスト項目に対して、テストデータとしては Web ブラウザの画面への入力値、事前にデータベースへ投入しておくデータといったものが必要となり、これらもテスト設計の段階できちんと作成しておく必要がある。

3. 技術課題の議論

本節では、目指す世界を実現するための技術課題について議論する。

3.1 ソフトウェア設計書の記法

ソフトウェア設計書はテストを生成するためのモデルを構築するための元情報として用いられる。モデルの情報量が多ければ、テストで必要となる情報も多く生成できるようになるが、設計書の記法を複雑にすると、開発現場への導入障壁が上がってしまうのが難し

^{†1} NTT ソフトウェアイノベーションセンタ
NTT Software Innovation Center

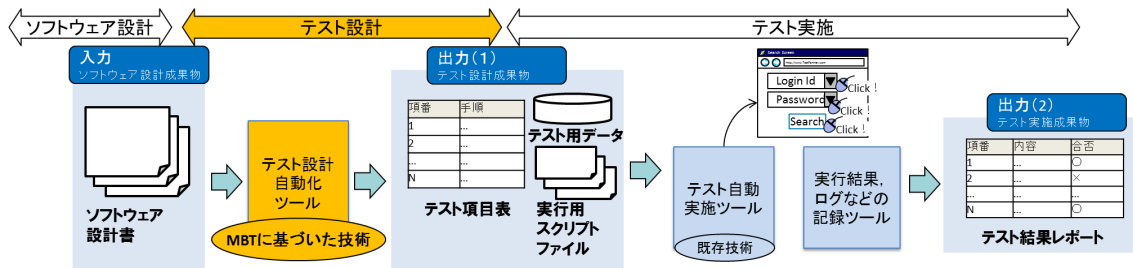


図 1 本研究の目指す世界

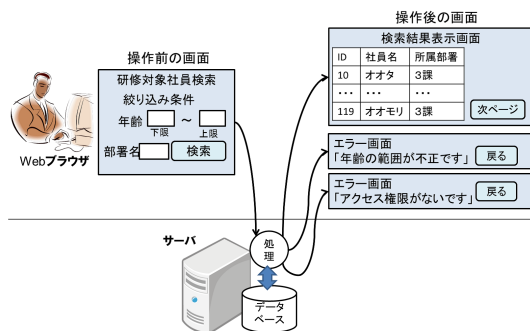


図 2 対象とする結合テスト

い点である．提案するアプローチでは，開発現場で実際に使用されている設計書のフォーマットに近い記法を採用することで，導入障壁を下げている．しかしながら，画面入力値や，データベースの各テーブルのレコードなど，実際にテストを実施するときに必要なテストデータ的具体値を生成しようとすると，どうしてもより詳細な設計情報が必要となり，より複雑な記法が必要となってしまう．そのため，開発現場への導入を進めるにあたっては，段階的に設計書の記法に慣れてもらうと共に，設計書の記述コストに見合うだけの嬉しさ（テストのコスト削減効果）を説明していくことが重要と考えられる．

3.2 テストデータ生成

業務システムの結合テストでは，画面入力値（文字列型，日時型を含む）やデータベースのレコードなど，複雑な構造をもったテストデータを生成する必要がある．このようなテストデータ生成の技術は主にソースコードを対象とした単体テストにおいて，これまで多くの研究がなされている．単体テスト，結合テストのどちらにしても，技術的には，意図したテストを行うために，特定の条件を満たすようなテストデータ的具体値を生成するため，結合テストのテストデータ生成においても，単体テストで培われたテストデータ生成の技術は応用可能であると考えられる．例えば，筆

者らは過去にソフトウェアの設計モデルから，シミュレーション用のソースコードを生成し，そのソースコードに対して，主に単体テストで用いられる技術である Concolic Testing を適用することで，画面入力値，データベースの具体値を生成する手法 [1] を提案している．

3.3 テストの合否判定

テスト全自動化のために，合否判定の自動化は必要不可欠な技術であるが，一般に Test Oracle 問題として知られ，自動導出が難しい領域とされている．提案するアプローチは MBT に基づいており，ソフトウェアの設計モデルが入力として存在するため，テストの結果としてどの画面に遷移するか，テストの事後条件として画面を構成する各要素の状態がどのようになっていけばよいかなどなど，合否判定に必要な情報をある程度設計モデルから抽出できる可能性がある．もちろん，画面のレイアウト崩れの確認など，自動では難しい領域もあるため，機械で確認する領域と人手で確認する領域を明確にすることが重要であろう．

4. おわりに

業務システムの結合テストをスコープとし，MBT の考え方に基いて，ソフトウェア設計書からテスト項目，テストデータ，テストスクリプトを網羅的に自動生成するアプローチを紹介し，技術課題についての議論を行った．目指す世界の実現に向け，開発現場の声も参考にしながら今後も技術検討を進めていきたい．

参考文献

- 1) 丹野治門，張曉晶，生沼守英．ソースコード生成を利用した結合テスト向けデータベース生成手法の提案．情報処理学会研究報告ソフトウェア工学，Vol. 2014-SE-185，No.25，pp. 1-8，jul 2014.
- 2) 丹野治門，張曉晶，田端啓一，生沼守英，村主一仁．ソフトウェアの品質確保と開発コスト削減を目指したテスト自動化技術．NTT 技術ジャーナル，Vol.25，No.10，pp. 19-22，oct 2013.