# 実行時情報を用いた機能抽出によるシステムの実装と要求との結び付け

渡邊　結[†1]　石尾　隆[†1]　井上克郎[†1]

本研究ではシステムの保守作業を支援するために，オブジェクト指向プログラムによって実装されたソフトウェアシステムに対して，システムの実現する要求，プログラムの持つ機能，機能の実装との間の一連の繋がりを復元する．システムへの要求が 1 つのテストケースシナリオで表現されるとみなし，その実行時情報からプログラムの持つ機能を自動抽出することで，要求仕様・機能設計・詳細設計間の関連を検出，可視化する．その結果，保守作業におけるプログラム理解を容易にし，各種保守作業のコストを軽減することを目指す．

# Extracting Features in Execution Traces for Linking Requirements and Implementation.

Yui Watanabe,[†1] Takashi Ishio[†1] and Katsuro Inoue[†1]

For program understanding on software maintenance, developers have to take on a difficult task to figure out traceability relationship among requirements, features and their implementation. We propose an automatic approach to extract these relationships from a current system. Our technique is detecting features in an execution trace of object-oriented program using consolidated test cases corresponding to each requirement.

## 1. ソフトウェアシステムの保守・理解支援

ソフトウェアシステムのライフサイクルにおいて最もコストがかかるのは保守段階であるが，特にプログラム理解作業がその半分以上を占めることから，担当者の大きな負担となっている．

保守担当者が既存のシステムに新たな要求を実現させるために機能の追加や変更を加えたり，運用中に発見されたバグの原因となる部分をプログラム中から特定するには，まず対象となるシステムの実現する要求と実現に利用されるプログラムの持つ機能，およびそれら機能の実装の詳細との間の関連を正しく理解していなければならない．プログラム理解に用いられる代表的な手法はソースコードの読解であるが，オブジェクト指向プログラムは実行時の環境やシステムの状態によって動作が決定するため，ソースコード上にはシステムの設計情報に関連する記述が含まれておらず，システムを正しく理解するためには仕様書などの設計情報を参照する必要がある．しかし，長期に渡って保守され続けたシステムや再利用のため外部から持ち込まれたプログラムなど，仕様書の内容と実際のシステムの実装に齟齬がある場合や，そもそも設計情報を十分に参照できない場合がある．このため，現状のシステムに対する正確な理解が困難となり，保守コストが増大してしまう．

†1 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

## 2. 設計情報と実装との関連の復元

本研究では，システムへの要求に対して，それを実現する機能及びその実装との間の一連の繋がりを復元する手法を提案する．保守対象システムの実行時情報を解析することにより現在のシステムに合致する機能設計情報を復元し，要求，実装間の対応付けを行う．要求，機能，実装の 3 段階でのシステム構成とその関連を可視化することで，保守担当者はシステムの要求から実装までの関連を容易に理解し，またこの対応関係を利用して変更の影響範囲を特定するなど，各種保守作業のコストを軽減することができる．

### 2.1 要求，機能，実装の粒度と関連

提案手法はオブジェクト指向プログラムによって実現されたシステムを対象としている．

システムへの要求は，設計段階においてシステムの要求仕様や外部仕様として定義されるものである．本研究ではプログラムの持つ複数の機能の実行によって実現される機能的要求を対象とし，システムの制限や品質など特定の機能の実行によって構成されない要求は原則として対象としない．本研究で対象とする要求は，システムの統合テストケースに対応するものとす

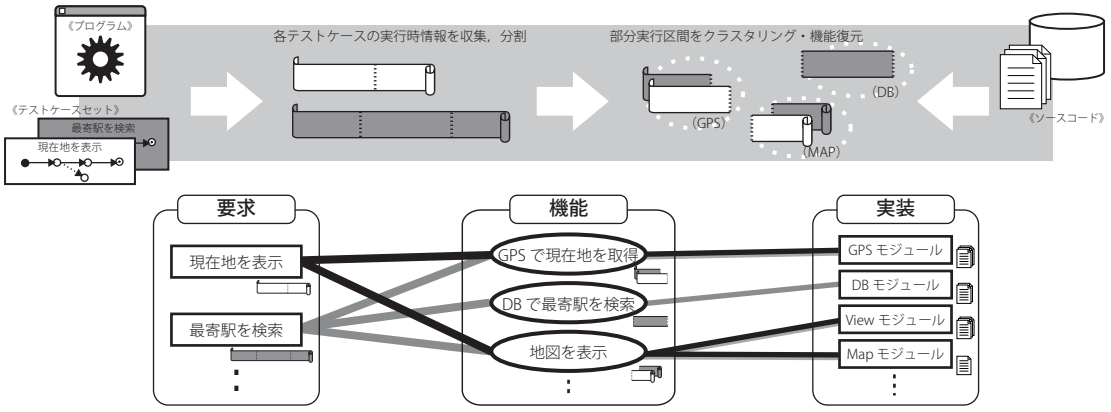


**図 1**　提案手法の概要：要求―機能―実装の関連図の例（部分）
Fig. 1　An example of relationship among requirements, features and implementation.

る．1 つのテストケースの実行が，1 つの要求の実現シナリオである．

プログラムの持つ機能は，設計段階においてシステムの内部仕様やプログラムの上位設計として定義されるものである．システムへの要求は，プログラムの持つ複数の機能の実行によって実現される．機能の粒度には様々な定義が可能であるが，本研究では，1 つの機能をテストケースシナリオ中の 1 ステップに対応するものとする．あるテストケースの実行時情報はある特定の実行区間ごとに分割することができ，それぞれの区間が何らかの機能の実行に対応する．

プログラムの実装は，設計段階において詳細設計として定義される．本手法では，ソースコード上の静的な構成から，各種保守作業の目的に応じた任意の粒度のモジュールと機能の対応付けを行う．

### 2.2　手法の概要

提案手法は，システムの実現する要求に対応するテストケース群とプログラムを入力として，プログラムの持つ機能と要求，実装との関連を出力する．**図 1** 上部に提案手法の概要，下部に出力される関連図の一例を示す．

まず，入力された全てのテストケースを実行して実行時情報を取得し，それぞれ 1 機能の実行区間ごとに分割する．これは，テストケースシナリオの情報を入力として，我々の提案しているフェイズ自動検出手法[1] を用いて実現する．

次に，プログラムの持つ機能を復元する（図 1 下部で楕円で示された機能ノードに対応）．分割された部分実行区間の集合を，1 つのクラスタが 1 つの機能に対応するようにクラスタリングする．同じ機能を実行した区間においては，動作したオブジェクトや発生したイベントは完全には一致しないが，それらに対応するソースコード上の記述範囲はある程度共通しているはずである．したがって，各区間で実行されたイベント集合の持つ静的情報の類似度によるクラスタリングを行う．また，各クラスタには適切な機能名を与える必要がある．現在は完全な機能名を自動生成することが困難であるため，実行されたメソッドやそのクラス名のうち，そのクラスタに特徴的なものを複数選出することで利用者に機能名を推測させるアプローチを試行している．ただし，入力として与えられるテストケースのシナリオ中に機能の名前に関する情報が含まれている場合，テストケースと部分実行区間の対応関係から直接機能名を割り当てる，あるいは実行区間からより適切な名前を選出できると考えられる．

最後に，テストケースとその実行時情報を分断した各実行区間が属するクラスタとの対応から，要求と機能との関連を抽出する．また，各クラスタに属する各実行区間内で呼び出されたモジュールとの対応から，機能と実装の対応を抽出しすることで，要求-機能-実装間の関連を復元する（図 1 下部におけるエッジに対応）．

今後，クラスタリングと名付けのアルゴリズムを確立し，提案手法を実装し実際のシステムに対する適用実験を行う予定である．

**謝辞**　本研究は，文部科学省グローバル COE プログラム（研究拠点形成費）の補助によるものである．

## 参　考　文　献

1) Watanabe, Y., Ishio, T. and Inoue, K.: Feature-level Phase Detection for Execution Trace Using Object Cache, *Proceedings of the 6th International Workshop on Dynamic Analysis*, pp.8–14 (2008).