

CRUD マトリクスを用いたソフトウェア設計影響分析手法

加藤 正恭[†] 小川 秀人[†]

[†]株式会社日立製作所 中央研究所

1. はじめに

ソフトウェア派生開発においては、新機能の追加が既存機能の動作に悪影響を与えないよう、設計時にどこにどのような影響が及ぶかを分析しておかなければならない。

性能要求の高い組込みソフトウェア開発では、処理速度を向上する実装上の工夫としてあえてグローバル変数を用いて開発することがある。このような製品では、グローバル変数を介した処理間の連係動作が見えにくくなる。よって、ソフトウェアを変更した場合の影響が特定しにくいという課題がある。

従来は、影響が波及しそうな機能に目安をつけて絞込み、人手で仕様書やソースコードを調査して影響を分析していた。しかし、大規模なソフトウェアの場合は、人手では限界がある。

本研究では、ソースコードから関数と変数の依存関係に関連する情報を自動的に抽出して、CRUD マトリクス[1]として可視化する手法を開発した。本手法により、グローバル変数を介した相互作用が多いソフトウェアにおいても、効率的に影響分析ができるようになった。

2. CRUD マトリクスによる設計影響分析

2.1. CRUD マトリクス

CRUD マトリクスとは、機能とデータを2軸にもつマトリクスであり、機能によるデータアクセス仕様を可視化したもの(図1)である。

CRUDマトリクス

	データ1	データ2	データ3
機能A	U	C	
機能B		RU	U
機能C	U	D	R

C:作成(Create)
R:参照(Read)
U:更新(Update)
D:削除(Delete)

"機能Bがデータ3を更新すること"をあらわす

図1 CRUD マトリクス

従来、CRUD マトリクスは業務システムの開発でデータ設計分析に用いられてきた。

本研究では、機能に関数に、データを変数に置き換えることによって、関数による変数アクセス仕様を表形式で可視化し、関数と変数の相互作用を分析する手法について検討した。

2.2. CRUD マトリクスによる設計影響分析

CRUD マトリクスによる設計影響分析プロセス(図2)について述べる。まず、派生開発のベースとなる既存ソースコードから CRUD マトリクスを生成(①)する。次に、機能の追加・変更に関する情報をもとに CRUD マトリクスを修正(②)する。そのあと、CRUD マトリクスから設計リスクを抽出し、適宜設計を見直す(③)。これらの作業を設計リスクがなくなるまで繰り返し、機能追加・変更の影響を洗い出す。抽出可能な設計リスクには、例えば「従来は特定の関数でしか更新できなかった変数が、今回追加した関数でも更新可能になったので、従来の更新関数が誤動作しないか」などがある。

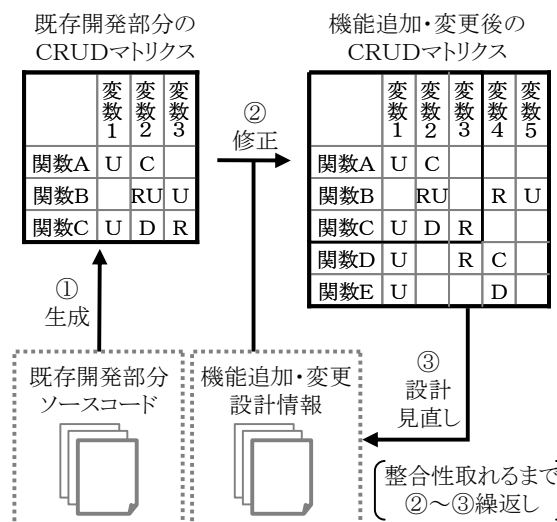


図2 CRUD マトリクスによる影響分析プロセス

また、実装完了後に改めてソースコードから CRUD マトリクスを生成し、設計時のものと比較することで、設計どおり実装されたかどうかの確認もできる。

Software Impact Analysis Method Using CRUD Matrix

Tadahisa Kato[†], Hideto Ogawa[†]

[†] Central Research Laboratory, Hitachi, Ltd.

3. CRUD マトリックスの生成と拡張

ここでは、既存のソースコードから CRUD マトリックスを生成する方法、および、生成した CRUD マトリックスを実用的サイズに縮小する方法について述べる。

3.1. CRUD マトリックスの生成

まず、既存ソースコードの静的解析を行い、関数リスト、変数リスト、および変数参照情報(どの関数がどの変数を参照または更新するか)を抽出する。抽出したデータを表形式で可視化すると CRUD マトリックスになる。関数による変数の参照を R、更新を U として記す。また、リストなど可変個の要素をとるデータ構造では、リストへの要素の追加を、リスト自身の更新(U)および追加する要素の作成(C)として記す。

リストからの要素の削除も同様に U、D と記す。

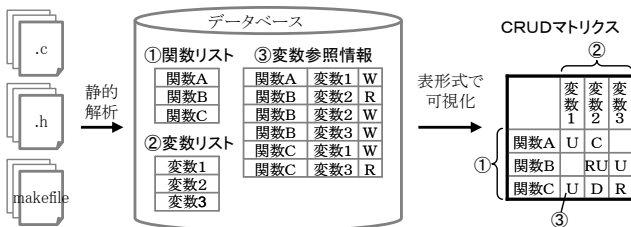


図 3 CRUD マトリックスの生成

3.2. CRUD マトリックスの抽象化

CRUD マトリックスは、行や列を統合することでサイズを縮小できる。関数軸方向は、呼び出し元関数に呼び出される関数の CRUD 情報を統合する。また、変数軸方向は、構造体にそのメンバの CRUD 情報を統合する。これらの処理は静的解析結果を用いれば自動化できる。

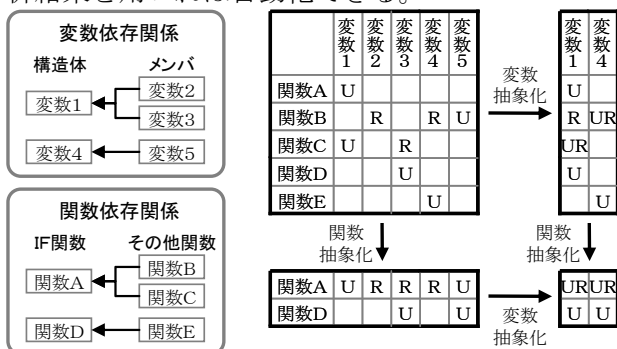


図 4 CRUD マトリックスの抽象化

3.3. CRUD マトリックスの絞込み

CRUD マトリックスは、分析に不要な関数や変数を除外することでサイズを縮小できる。あらかじめ影響分析の基点となる関数または変数を指定し、それらに変数アクセス上関連のある関数および変数のみを抽出して CRUD マトリックスを生成する。

図 5 の例で説明する。予め基点関数に関数 B を指定したとする。まず、基点関数(関数 B)がアクセスする変数(変数 3, 変数 5)を基点変数に追加する。次に、追加した変数にアクセスする関数のうち基点関数に含まれないもの(関数 D)を追加する。これらに関数と変数が追加されなくなるまで繰り返す。得られた基点関数と基点変数に絞って CRUD マトリックスを生成すれば、予め指定した関数や変数に関連する範囲に限定した CRUD マトリックスを生成できる。

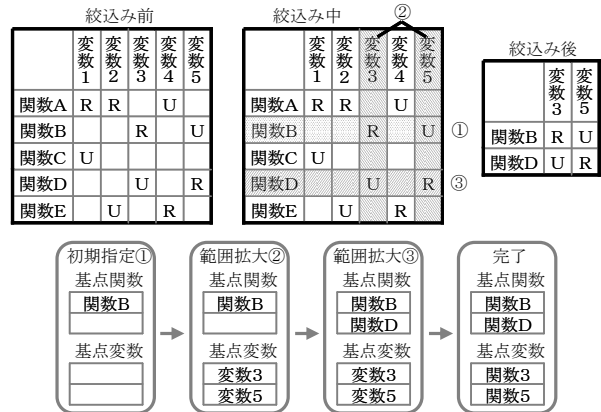


図 5 CRUD マトリックスの絞込み

4. 評価

本手法を 100 万行規模のソフトウェア開発の影響分析に適用し、工数削減効果を評価した。従来も CRUD マトリックスによる影響分析を実施していたが、手作業中心のため、機能あたり 1ヶ月の準備工数を要していた。本手法を適用し作業の一部を自動化することで、影響分析の準備工数を 3 時間(従来比 1/50)まで削減した。

5. 結論

本研究では CRUD マトリックスによるソフトウェアの設計影響分析手法を開発した。ソースコードから CRUD マトリックスを生成することで影響分析を効率化した。また、CRUD マトリックスの抽象化や絞込みにより、CRUD マトリックスを実用的なサイズに纏められるようにした。一方、情報源にソースコードを用いたため、CRUD マトリックス上の語句が実装レベルの表現になり分かりにくくなった。より平易な表現への置き換え方法については今後検討する。

参考文献

[1] H. Kilov, "From semantic to object-oriented data modeling", *First International Conference on System Integration*, 1990. 385 - 393.