

# データ結合と制御結合を満たすことについて

## Meet data coupling and control coupling

### 背景

ソフトウェア工学の黎明期以来、ずっと唱えられてきた概念の一つがモジュール化です。これは構造化設計と機能分割によるアプローチで、ソフトウェアを明確に定義された機能単位、すなわち曖昧さのないインターフェースを持ったモジュールに分解するものです。このアプローチによって高品質なソフトウェアが生み出されることが実証されています。モジュール化されたセーフティクリティカルなシステムにおいては、モジュールがどのように一体となって相互作用しているかを調べて、データ結合や制御結合に欠陥がないことを確認する必要があります。

### 凝集度 (Cohesion)

データ結合と制御結合を正しく理解するためには、まず「凝集度」を理解する必要があります。凝集度とはモジュラーなソフトウェアコンポーネントの性質です。よく定義されたインターフェースを持ち、他のモジュールから独立して動作するモジュールは、高い凝集度を持つと考えられます。コマンドラインの MD5 暗号的ハッシュジェネレータは、高い凝集度を示すソフトウェアコンポーネントの一例です。これの入力は任意に記述されたデータオブジェクト(ファイルなど)であり、出力は base64 によるハッシュ表現です。ハッシュジェネレータは単一の入力を受け取って単一の出力を提供するものであり、外部依存性を持たないため高い凝集度を示すと考えられます。

凝集度の高低のもう一方の端には飛行制御システムがあります。これは、飛行機の現在の状態(たとえば、地上にいるかどうか)や、入力される飛行データストリーム(対気速度、高度、姿勢など)に大きく依存するため、凝集度が低いものです。

あるシステムからソフトウェアモジュールを抽出して別のシステムで再利用しようとする、凝集度によって大きな影響を受けることになります。思うように行かない隠れた依存性との闘いについて誰もが喜んで話してくれるでしょう。そしてここに問題があるのです。もしモジュールが真に凝集していないのであれば、各モジュールをその依存者すべてによるコンテキストでテストするのはどうすれば可能でしょうか？ これを実現するにはソフトウェアモジュール間に存在する結合度を理解する必要があります。結合度は、ソフトウェアの凝集度を測定する手段を提供し、ソフトウェアの品質を評価・改善するために使用できるメトリクスも提供します。

### 結合度 (Coupling)

結合度の概念は、1980 年に Meilir Page-Jones が「The Practical Guide to Structured Systems Design」で、「あるモジュールが他のモジュールに依存する度合い、具体的には、あるモジュールの欠陥が他のモジュールの欠陥として現れる可能性の尺度、あるいは、あるモジュールの変更によって他のモジュールの変更が必要になる可能性の尺度」と定義しました。

Page-Jones は、結合度には基本的な種類が二つあるとし、それらを次のように定義しました。

- 制御結合 — あるモジュールが、別のモジュールに影響を与えるという明示的な目的のために、その別モジュールに情報を伝達するタイプのもの
- データ結合 — あるモジュールが別のモジュールに情報を伝達するタイプのもの

両者の違いを明確にするために例を見てみましょう。

航空機のエンジン制御システムは、高度や対気速度などの航空データ情報を利用しますが、航空機が地上にあるときと空中にあるときとは、使用する制御アルゴリズムが違います。どのアルゴリズムを使用するかは、航空機が地上にあるかどうかを示す着陸装置システムからの「車輪荷重」信号によって決定されます。この例では、エンジン制御システムが高度と対気速度のデータストリームに依存していることから、それらを提供するシステムとのデータ結合度が高いことがわかります。同様に「車輪荷重」信号は、エンジン制御システムと着陸装置システムの高い制御結合度を示しています。

結合度メトリクスを使用して、ソフトウェアの設計と検証のプロセス全体を通じてソフトウェアの品質を向上させることができます。その目的は、ソフトウェア設計で意図された方法でのみソフトウェアモジュールが互いに影響し合うことを示し、予定外の動作や異常な動作、誤った動作がないことを保証することです。設計時にデータ結合と制御結合を文書化することで、ソフトウェア統合プロセスにおいてテストするための一連の要件が得られます。同様に、モジュール間のデータ結合と制御結合が確実に実行され、ソフトウェアテスト中に障害がないことを確認することは、ソフトウェアの統合とアーキテクチャが完全に検証されていることを示します。

DO-178C では、これらの設計・統合・テストの目的が満たされていることを確認するために、セーフティクリティカルなソフトウェアに対して制御結合とデータ結合の評価を実施することを要求しています。制御結合とデータ結合の測定は、制御フロー解析とデータフロー解析を組み合わせることで実現されます。

制御フロー解析は、プログラムの呼び出し階層と個々の関数の両方で実行されます。

データフロー解析は、ソースコードを通して変数を追跡し、異常な使用があれば報告します。このチェックは、関数のレベルでもシステム全体での解析の一部としても実行されます。これは非常に強力な手法であり、データ結合の評価の基礎を提供するだけでなく、初期化される前に使用される変数や、境界を越えてアクセスされる配列など、他の重大な問題も検出します。

その名前から、制御フロー解析が制御結合を可視化し、データフロー解析がデータ結合を可視化するものと思いがちですが、そうではありません。制御結合とデータ結合の両方を評価するためには、制御フロー解析とデータフロー解析の両方が必要です。

DO-178C による要求とは関係なく、制御結合とデータ結合を介してソフトウェアモジュールの凝集度を測定することは、変更による影響を軽減してソフトウェアの再利用を簡素化し、ソフトウェアのテスト容易性と保守性の改善に役立ちます。

## まとめ

よく定義されたインターフェースを持ち他のモジュールから独立して動作するモジュールは、凝集度が高いと考えられます。制御結合とデータ結合のメトリクスは凝集度の尺度であり、どちらも制御フロー解析とデータフロー解析の両方を適用することで導出できます。

結合度メトリクスを使用して、ソフトウェア設計と検証のプロセス全体を通じてソフトウェアの品質を向上させることができます。その目的は、ソフトウェア設計で意図された方法でのみソフトウェアモジュールが互いに影響し合うことを示し、予定外の動作や異常な動作、誤った動作がないことを保証することです。



富士設備工業株式会社 電子機器事業部 [www.fuji-setsu.co.jp](http://www.fuji-setsu.co.jp)

© LDRA 株式会社 このドキュメントは LDRA Ltd. の所有物です。  
その内容を当社の承諾なく複製、開示、利用することはできません。