ソフトウェア工学

04: 要求分析／定義と設計(Ⅱ)
－ 構造化設計

理工学部 経営システム工学科
庄司　裕子

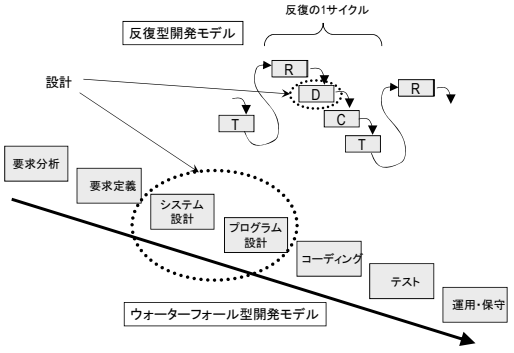
今回のテーマ

⇨要求分析／定義と設計(Ⅱ)
－ 構造化設計 (←分析は前回学習)
- 開発プロセスにおける位置づけと目的
- モジュール構成図
- モジュール構成の良否の目安
- モジュール分割の手法

要求分析／定義と設計(Ⅱ)
－ 構造化設計

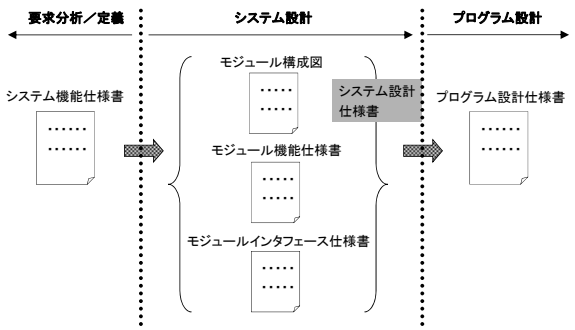
⇨開発プロセスにおける位置づけと目的
- モジュール構成図
- モジュール構成の良否の目安
- モジュール分割の手法

開発プロセスにおける位置づけ



設計の目的

- 対象システムをどう作るか(how)を決定する



システム設計とは

- 設計者(またはSE)が
- システム機能仕様書を基に
- システムのモジュール構成、個々のモジュールの機能、モジュール間のインタフェースを決定し
- システム設計仕様書(モジュール構成図、モジュール機能仕様書、モジュールインタフェース仕様書)にまとめる

プログラム設計とは

- プログラマが
- システム設計仕様書を基に
- 個々のモジュールに使用するアルゴリズム（ロジック）とデータ構造を決定し
- プログラム設計仕様書（あるいはロジック設計仕様書）にまとめる

要求分析／定義と設計（Ⅱ）－ 構造化設計

✓開発プロセスにおける位置づけと目的
⇨モジュール構成図
- モジュール構成の良否の目安
- モジュール分割の手法

構造化設計（Structured Design：SD）とは

- システム設計
  - 目的とするシステムをどのようなモジュールの組み合わせで実現するかを決定

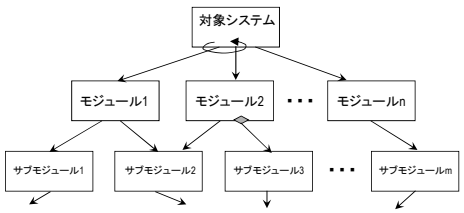
（その結果をシステム設計仕様書に記述）
⇨そのためのモジュール分割の指針を与える手法が構造化設計
※段階的・階層的に分割していくのが「構造化」設計

システム設計仕様書

- モジュール構成図
  - モジュール群をどのように構成するかを規定
- モジュール機能仕様書
  - 各モジュールの機能を規定
- モジュールインタフェース仕様書
  - 各モジュール間のインタフェースを規定

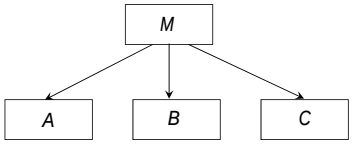
モジュール構成図

- モジュール間の呼び出し関係を表す図
  - 呼び出し元（caller）と呼び出し先（callee）
  - ルートとリーフを除き、中間モジュールは呼び出し元にも呼び出し先にもなる



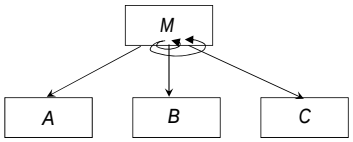
モジュール呼び出しの基本形（Ⅰ）

- 連接構造
  - ☞ *A*、*B*、*C*が*M*の下請けとして呼び出される



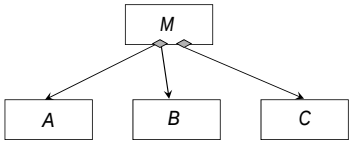
モジュール呼び出しの基本形（Ⅱ）

- 反復構造
  - ☞*B*の反復呼び出しが生じ、さらにそれと*C*を含めた大きな反復呼び出しが生じる



モジュール呼び出しの基本形（Ⅲ）

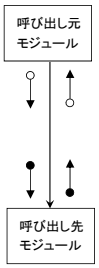
- 選択構造
  - ☞ある条件によって*A*と*B*の選択的呼び出しが生じ、別の条件によって*C*の選択的呼び出しが生じる



モジュール間でのパラメータの受け渡し

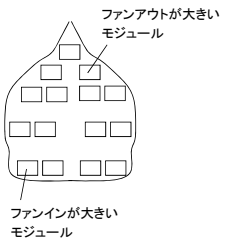
- 白丸はデータパラメータ
- 黒丸は制御パラメータ
- 矢印は受け渡しの方向

⇨モジュール構成図を見れば、パラメータに関する情報はすぐわかる



モジュール構成の良否の目安（Ⅰ）

- 一般にモスク型の木構造が良い
  - 頂上近くでは比較的多くのモジュールに枝分かれ
  - 中間では下請けがないか少ないモジュールが増えるため、同一レベルでのモジュール数はほぼ一定
  - 末端では共有モジュールが増えるため、同一レベルでのモジュール数は中間レベルよりやや少ない



モジュール構成の良否の目安（Ⅱ）

- 各モジュールのファンアウト
  - 7個前後（3～9）がよいとされる（マジックナンバー7）
- 各モジュールのファンイン
  - モジュール構成が厳密な木構造であれば、ルート以外のすべてのモジュールはファンインが1
  - 共有モジュールについては2以上
  - 共有モジュールは木構造の上位には現われないほうがベター（変更時の影響が広範囲に及ぶため）

モジュール構成の良否の目安（Ⅲ）

- モジュールの規模（スケール）
  - ⇨ 一般に、扱いやすいモジュールの規模はソースプログラムで1ページ（50～60行）以内とされる

ただし、

- 規模だけで判断するのは避け、モジュールの結合度や凝集度を考慮して総合的に判断する

モジュールの結合度（coupling）

- モジュール構成図に現われるすべてのモジュールの対（ペア）について、モジュール同士の関連性を6種類に分類

良い ① 無結合：互いにまったく無関係
② データ結合：データパラメータ渡しによる直接呼び出し
③ 制御結合：制御パラメータ渡しによる直接呼び出し
④ 共通結合：共通データ領域を介した情報交換
⑤ 外部結合：外部の公開データ領域を介した情報交換
悪い ⑥ 内容結合：一方から他方の内容にアクセス

☞システム内のモジュール対の結合度の分布状況で、システム全体のモジュール構成の良否を判断

モジュールの凝集度（cohesion）

- モジュール構成図に現われる全モジュールについて、モジュールとしてのまとまりの良さを8段階に分類

悪い ① 偶発的：明確な理由なく恣意的にまとめる
② 論理的：見かけ上同一だが実際には異なる機能をまとめる
③ 時間的：実行する時間が近い処理をまとめる
④ 手続き的：一連の手続きをまとめる
⑤ 通信的：同一データを入力／出力する処理をまとめる
⑥ 逐次的：パイプライン的な処理をまとめる
良い ⑦ 機能的：明確に定義できる特定の機能のみをまとめる
⑧ 情報的：特定のデータへのアクセスをひとめとめにする

☞システム内のモジュールの凝集度の分布状況で、システム全体のモジュール構成の良否を判断

モジュール凝集度の判定

- モジュールが個々の凝集度の性質を満たすかどうか調べ、消去法で残ったものを割り当てる
- 複数の凝集度の性質を満たしているモジュールには、望ましくないほうの凝集度を割り当てる

モジュール分割の一般的指針（Ⅰ）

- 作成したモジュール構成に問題がある場合、手直しよりも最初からやり直すほうがよい結果が出る。
  - それまでの失敗を生かすことができる
- モジュール分割には十分時間をかける
- 得られた各モジュール構成の「良さ」を、いくつかの目安に基づいて評価し、メリット/デメリットを比較する

要求分析／定義と設計（Ⅰ）－ 構造化設計

✓開発プロセスにおける位置づけと目的
✓モジュール構成図
✓モジュール構成の良否の目安
⇨モジュール分割の手法

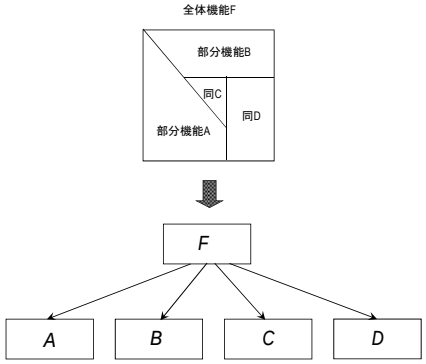
モジュール分割の手法

- 処理機能に着目する方法
  - 機能分割
- データの流れに着目する方法
  - STS分割
- ソフトウェアのパターンに着目する方法
  - トランザクション分割
- 複数の手法の組み合わせ
  - 複合分割

機能分割

① 全体をひとまとまりの機能とみなし、それを部分機能に分割する。これを詳細化と呼ぶ。
② それぞれの部分機能をさらに詳細化し、以下同様の作業を繰り返す。
③ 機能が十分に詳細になり、それ以上分割しなくても小さいプログラムとして実現できるようになったら、分割をやめる。

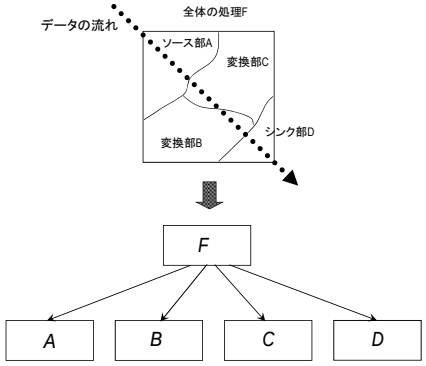
☞ 構造的な抽象化と階層化の考え方を直接、モジュール分割に適用したもの
☞ 処理機構が複雑なシステムに適している



STS分割

① 全体のデータの流れを、データの入力(ソース)、データの変換、データの出力(シンク)とみなして、それぞれの境界でモジュールに分割する。
② 分割した各モジュールをさらにソース部、変換部、シンク部に分割し、以下同様の作業を繰り返す。
③ 各モジュールが十分に詳細になり、それ以上分割しなくても小さいプログラムとして実現できるようになったら、分割をやめる。

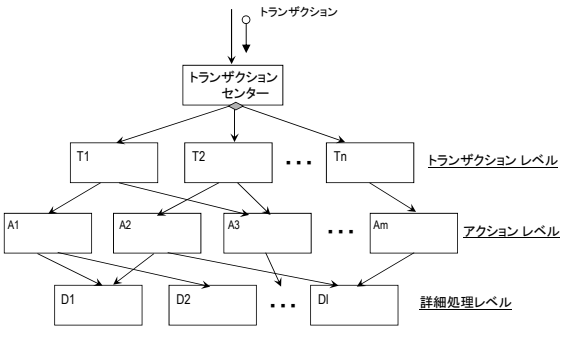
☞ データがどこでその性質を変えるかを見極めることが重要
☞ 事務処理システムなどのデータ指向システムに適している



トランザクション分割

① トランザクションごとの仕分けを行うディスパッチ モジュール(トランザクション センター)を1段目に置く
② それぞれのトランザクション タイプの処理を担当するトランザクション処理モジュールを2段目に置く
③ トランザクション処理モジュールの下請けをするアクション処理モジュール(多くは共有モジュール)を3段目に置く
④ 入出力などのより基本的な詳細処理モジュール(多くは共有モジュール)を置く

☞ 事務処理システムのうち、種々の伝票などの処理(トランザクション処理)に適している



複合分割

① 抽象度が高い初めの段階(モジュール構成図のルートに近い部分)ではSTS分割を用いる
② ある程度詳細化が進んだ段階(モジュール構成図のリーフに近い部分)で機能分割を用いる

☞ ヨードン法(変換分割法)
- DFDで記述されたシステム機能仕様書を直接利用して複合分割を系統的に行う方法

ヨードン法(変換分割法)

① DFDの階層図をモジュール分割の大まかな計画を立てる
② DFDの全体埋め込み図を見ながら、プロセスのグループ分けによるモジュール分割を行う
   A) 始めの段階ではデータの流れによって、ソース部、変換部、シンク部に分割する
   B) 詳細化が進んだら、機能分割に切り替える

☞ DFDがSTS分割にも機能分割にも有効な手がかりを与えてくれる

モジュール分割の一般的指針(Ⅱ)

- 変換分割を中心に、部分的にトランザクション分割など他の手法を取り入れる
- 完成したモジュール構成図はデータ パラメータの流れに応じて、以下に分類される
  - 入力モジュール: モジュール構成図の左側
  - 出力モジュール: 同図の右側
  - 管理(調整)モジュール: 同図のルート付近
  - 変換モジュール: 同図のリーフ付近

プログラム設計

- システム設計仕様書により、入力、出力、処理が明確に記述されている
- データ構造とアルゴリズムを選び、プログラム構造を決定する

☞構造化プログラミング(JSP法)
☞段階的詳細化法

まとめ:構造化設計

- システムの最適なモジュール構成を決定することが目的
- モジュール構成の良否の目安
- モジュール分割の手法
  - 処理機能に着目(機能分割)
  - データの流れに着目(STS分割)
  - ソフトウェアのパターンに着目(トランザクション分割)
  - 複数の手法の組み合わせ(変換分割)