

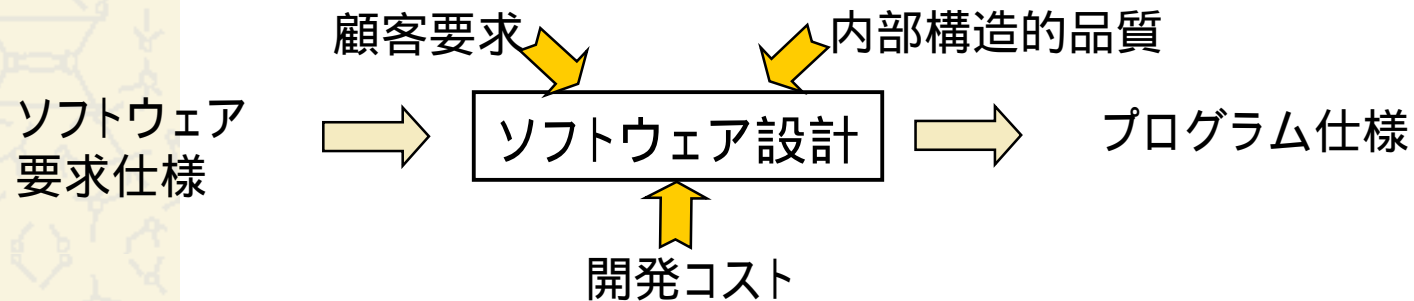


4章:ソフトウェア設計

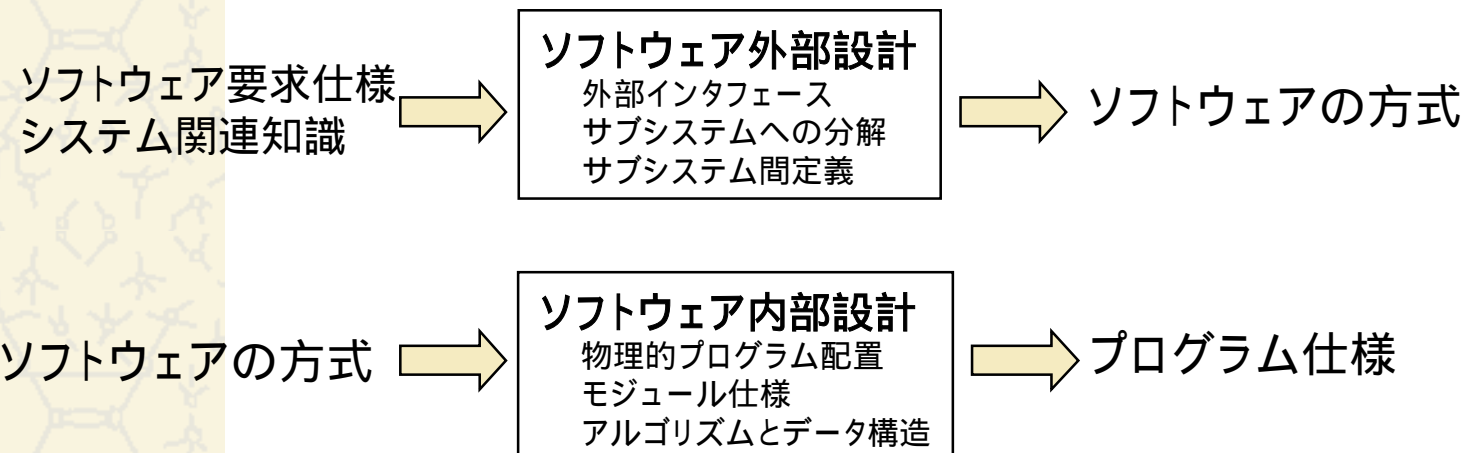
- 💡 ソフトウェア設計における基本事項
 - 設計とは
 - 設計の2つの段階
- 💡 ソフトウェア設計へのアプローチ
 - 良い設計
 - 3つの戦略
- 💡 モジュール分割
 - 複合設計法
 - データ構造分割
 - 共通機能分割
- 💡 モジュール分割の評価
 - モジュール間結合度
 - モジュール強度

4.1 ソフトウェア設計における基本事項

✧ ソフトウェア設計とその位置付け



✧ ソフトウェア設計の段階



4.2 ソフトウェア設計へのアプローチ

✳️ 良い設計: 要求品質を満たすソフトウェア構造を作る

- 品質: 信頼性・使用性・性能・安全性・セキュリティ・保守性
- コストと時間

✳️ 3つの戦略

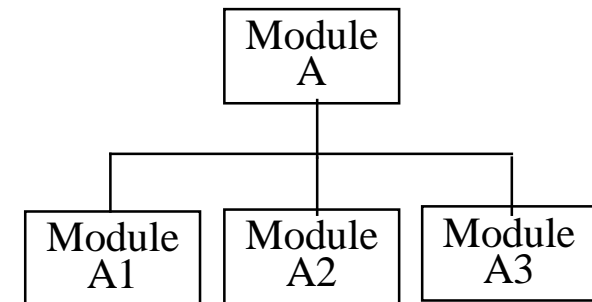
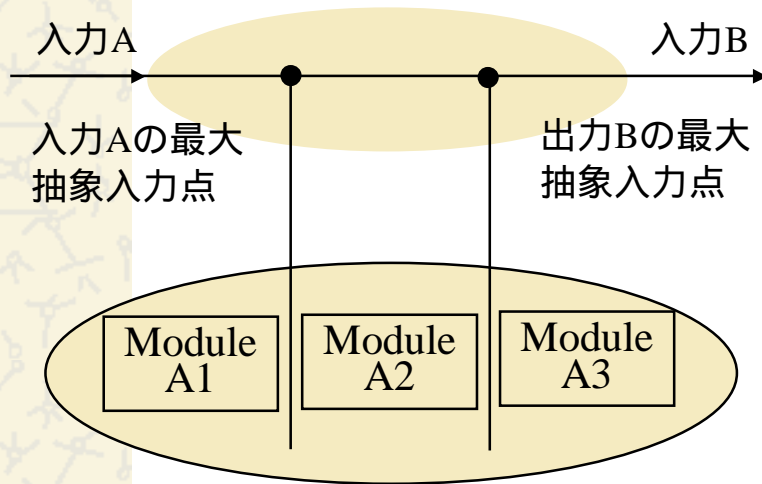
- 抽象化とモデルの利用: 状態遷移図, ペトリネット, HIPO
- 分割と階層化: システムは構成する要素数 N に対して2乗のオーダーで複雑さが増大するので, 複数の独立な部分に分ける
分割を再帰的にする 階層化される
階層化の場合の複雑度は, $N\log_N$ のオーダーとなるので, 複雑さの増大は防げる
- 独立性
 - 要素間には完全に独立になることはない
 - 要素内は密な結合だが, 他の要素との関係は疎かつ単純である

4.3 モジュール分割 / 複合設計法

💡 データの変換過程に着目したモジュール分解法

a. STS:Source Transform Sink

機能をデータの変換過程と考え、入力:処理:出力の3つの機能に分解する方法

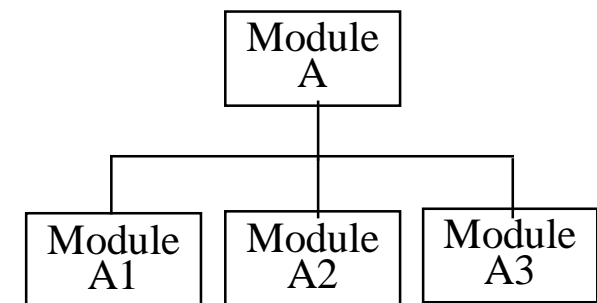
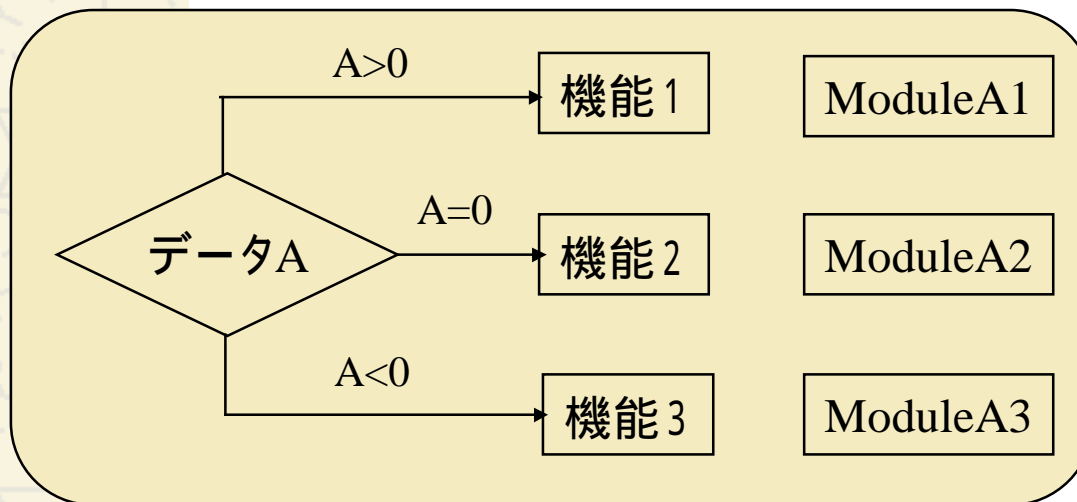


4.3 モジュール分割 / 複合設計法

💡 データの変換過程に着目したモジュール分解法

b. TR:Transaction分割

入力データに応じて処理が変わる場合に用いる分割方法

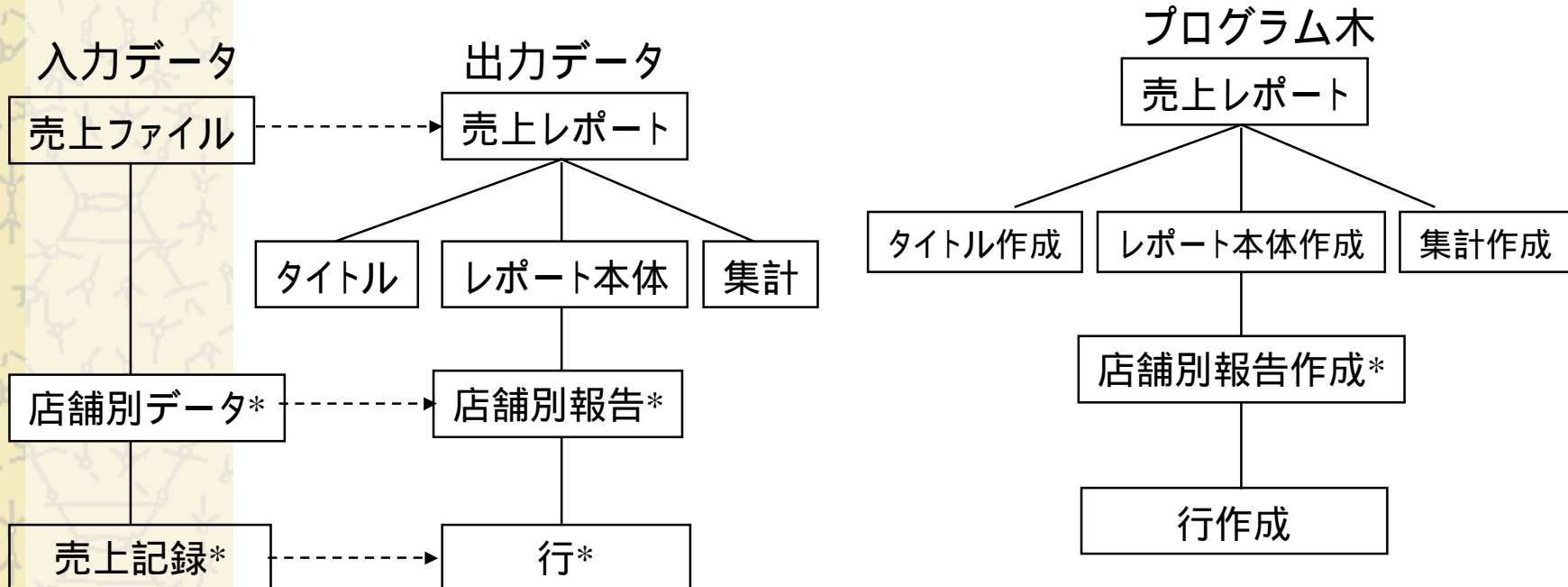


4.3 モジュール分割 / データ構造分割法

入力データ構造と出力データ構造に着目して処理の手順を検討する方法

・JSP:ジャクソン法によるモジュール分割の例

基本・繰返し・選択・連続の4つの構造要素





4.4 モジュール分割の評価

1. モジュールの大きさ

- ✦ 人間の認知範囲, プログラムコード100行ほど

2. 情報隠蔽の度合

- ✦ モジュール内部の設計事項を隠す

3. モジュール間結合度

- ✦ 2つのモジュール間の結合の強さ

4. モジュール強度

- ✦ 1つのモジュール内部の命令群がどれほど深く関わりあっているのかを示す



4.4 モジュール分割の評価

モジュール間結合度

モジュール間結合度の定義と順序関係

- a. 内容結合・・・他モジュール内の構文要素を直接参照, 変更する
- b. 共通結合・・・複数モジュール間でデータ領域を参照する
- c. 外部結合・・・モジュール内で外部参照可としたデータ領域を他モジュールが直接参照する
- d. 制御結合・・・他モジュールを呼出す際に, 相手モジュール内の制御を理解して, フラグやパラメータを渡す
- e. スタンプ結合・・・共有データ領域に無い同じ構造のデータ(構造体)を受け渡す
- f. データ結合・・・データの必要な部分だけ, 呼出しモジュールへの引数として渡す



4.4 モジュール分割の評価

モジュール強度

モジュール内の機能間の関連性

- a. 暗号的強度・・・モジュール間で機能の関連がない
- b. 論理的強度・・・関連する複数の機能を持つモジュールで制御変数により機能を選択
- c. 時間的強度・・・機能間関連は薄いですが、時間的順序関係に沿って起動される
- d. 手順的強度・・・各機能の起動には順序関係がある
- e. 連絡的強度・・・各機能の起動に順序があり、共通のデータを参照、または変更する
- f. 機能的強度・・・1つのモジュールで1つの機能を実現
- g. 情報的強度・・・同じ内部データを扱う機能を集め、個々のモジュールの機能は単一である(00のカプセル化の概念に近い)