

# アーキテクチャの計測と検証

FU14 ソフトウェア工学概論

第7回

吉岡 廉太郎

# 前回の内容

- システムが満たすべき品質要求も設計する
  - “品質観点”で評価する
  - 信頼性、堅牢性、セキュリティの対策が大きい
  - 様式の選択にも影響する

# 今日の内容

- アーキテクチャの洗練
  - 反復的に設計を良くする
  - 一つ以上のスタイル・様式を検討することが重要
- アーキテクチャ設計書(基本設計書)
- アーキテクチャの妥当性確認と検証

# アーキテクチャの評価と洗練

- 設計も反復作業
  - 設計上の判断を提案する
  - 評価する
  - 修正する
  - 再度、提案する
- アーキテクチャの評価手法
  - ①品質測定
  - ②トレードオフ分析
  - ③費用対効果分析

## ①品質測定

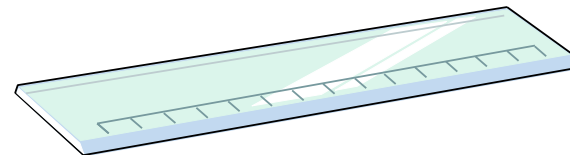
### • メトリックス

- あらかじめ定めた項目と定めた規準で数値化
- 数値のバランスで判断する
- 設計の「通知表」のようなもの

	a	b	c	d
値1	1	4	3	1
値2	4	1	3	4
値3	2	1	2	5
値4	2	5	1	1

### • 品質の主要素を評価するメトリックス

- オブジェクト指向システムのためのメトリックス
- 高レベル設計のメトリックス(結合度や強度など)
- ...



## ②トレードオフ分析

- 設計には複数の選択肢があることがほとんど
- 最良の設計を選択しなければならない
  - 最初に思いついた設計を単純に採用してはいけない
  - 開発メンバーが異なる設計を推すこともある
- 設計を比較する手法
  - 情報に基づいた選択をする
  - 妥当性を説明できるか

# 設計の比較評価

- 同じ課題を実現するのに、異なるアーキテクチャのスタイル(設計)を用いることができる
  - KWIC(Key Word in Context problem)問題
- 異なるアーキテクチャの4つの設計を考案した(Shaw and Garlan)

共有データ型

データ抽象化

暗黙的呼び出し

Pipe-Filter

## ②トレードオフ分析

# 例：KWIC索引システム

- 用途：高速検索（索引＝インデックス作成）
  - 図書館蔵書検索、ビデオライブラリの検索、Web検索、  
...
- 大量の文字列に対して、索引を作成するシステム
  - 入力された文を循環シフトし、全パターンをアルファベット順に出力する
  - 例）“Introduce Japanese culture to foreigners”の場合

入力

Introduce Japanese culture to foreigners

出力

culture to foreigners Introduce Japanese  
foreigners Introduce Japanese culture to  
Introduce Japanese culture to foreigners  
Japanese culture to foreigners Introduce  
to foreigners Introduce Japanese culture

### 全循環シフト

• Introduce Japanese culture

↓シフト

• Japanese culture Introduce

↓シフト

• culture Introduce Japanese

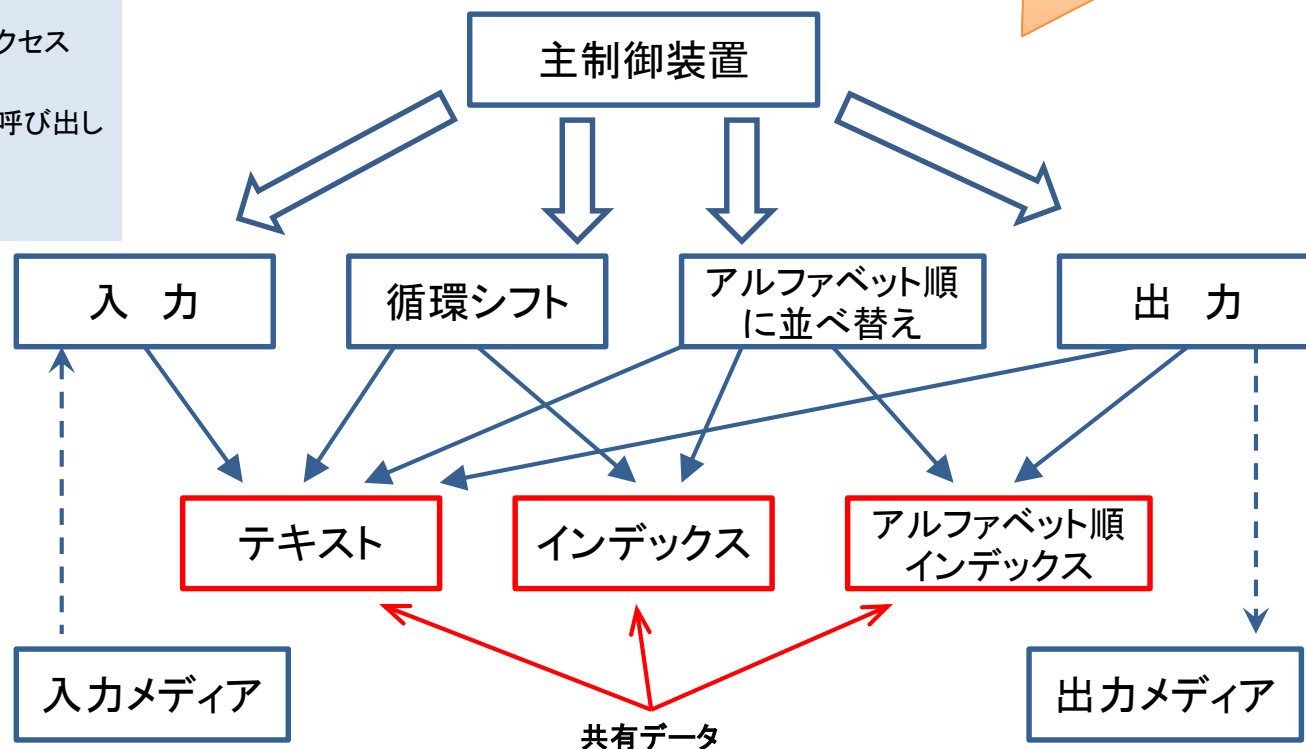


# KWIC:共有データ型

- 4つの機能ブロック
  - 入力、循環シフト、並び換え、出力

•効率が良い  
•データ形式を変えると全てを変更する必要がある

→ :ダイレクトメモリアクセス  
⇒ :サブプログラムの呼び出し  
--- :システム I/O

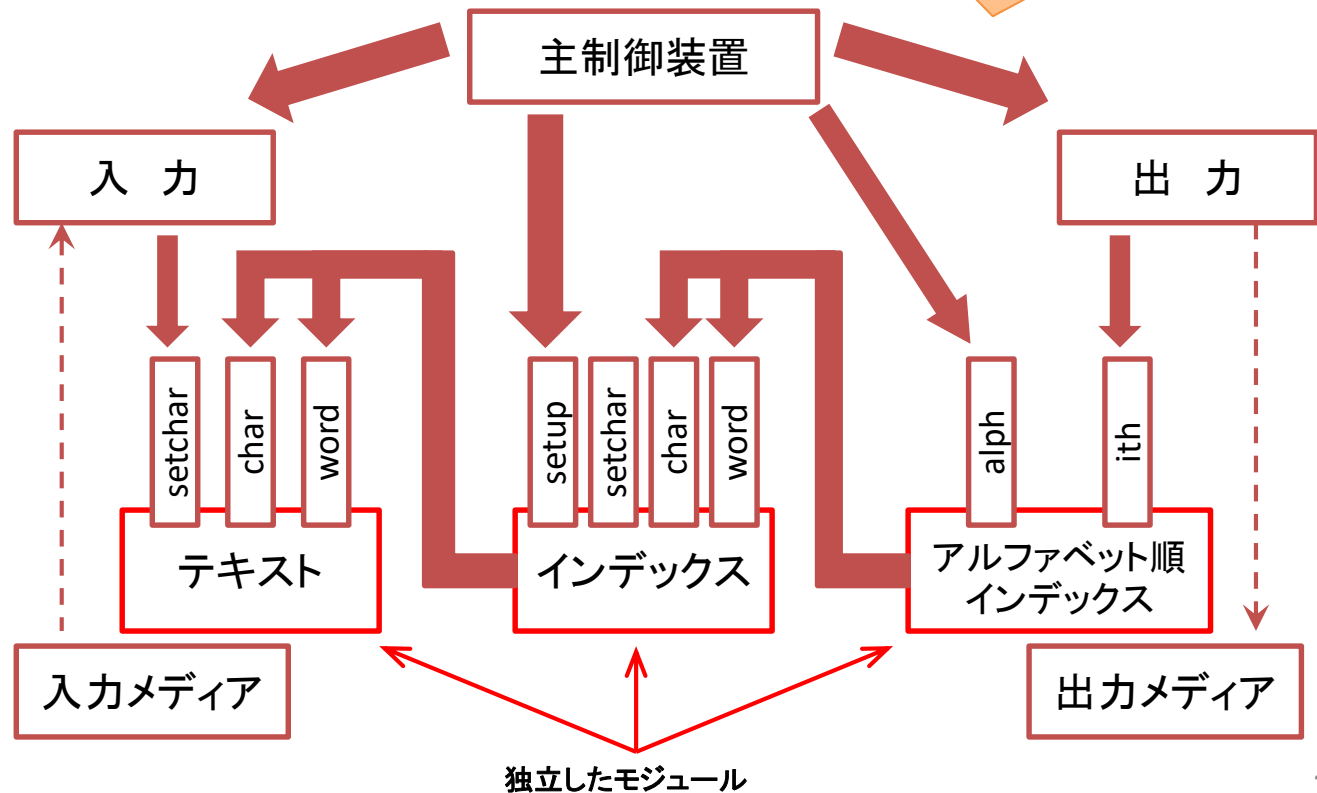


# KWIC:データ抽象化

- モジュールでデータを抽象化
  - データ形式を隠す

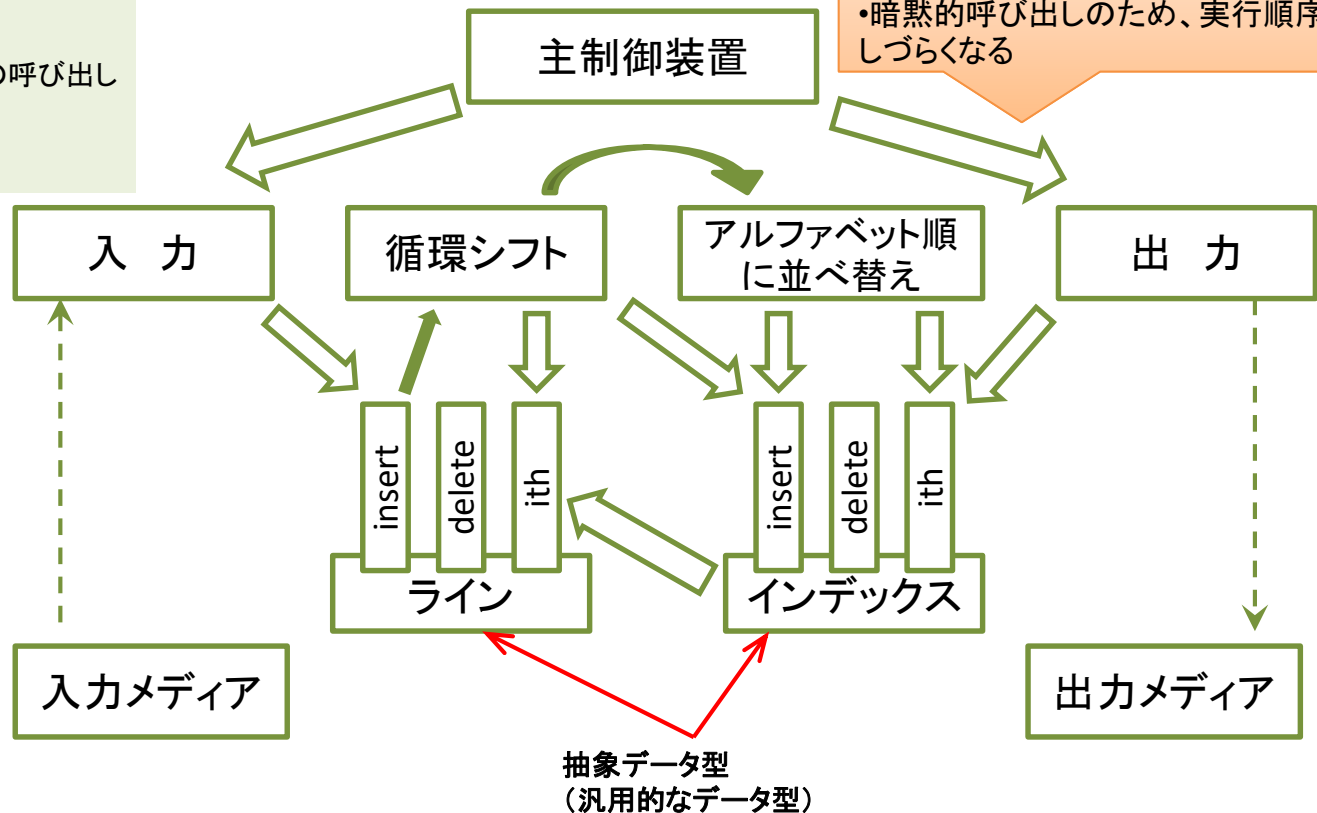
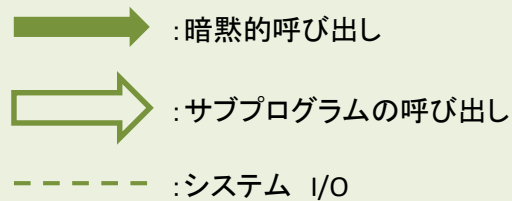
- データの抽象化により、データ変更に左右されずらい
- モジュールが再利用しやすい
- システムの機能変更が難しい

→ : サブプログラムの呼び出し  
- - - : システム I/O



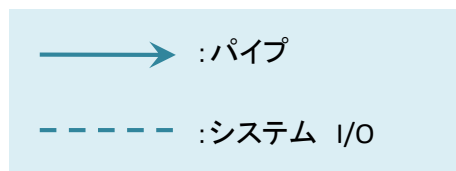
# KWIC:暗黙的呼出し型

- データ共有は無くなったが、分解は共有データ型と同様

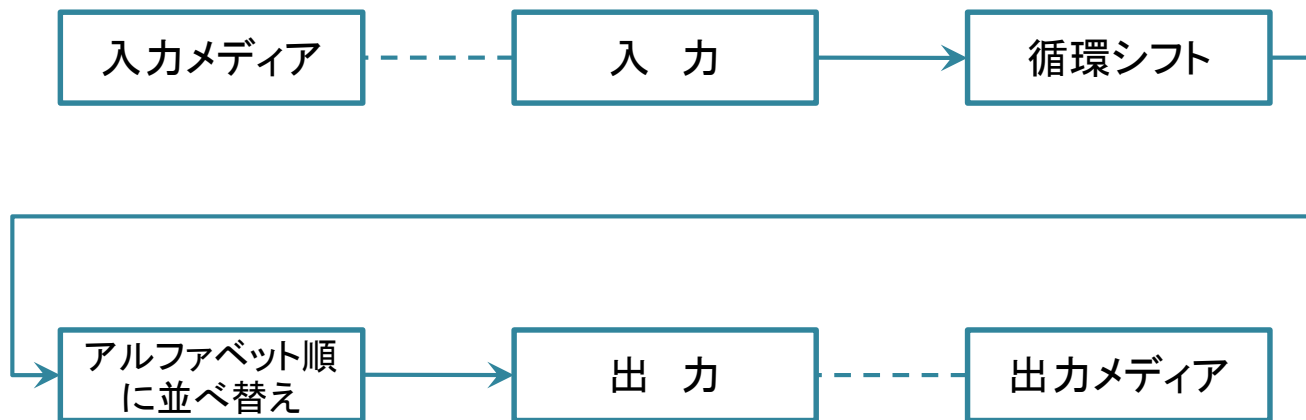


# KWIC: Pipe-Filter型

- 処理の順序はフィルタの順序として制御される



- 機能の追加が容易
- 機能変更が容易 (相互依存が少ない)
- 並列実行が可能
- 共有データを必要とする機能は不可能 (例、undo機能)
- 効率が悪い (データの転送が不可欠)



# 設計の比較評価

- 比較テーブルで対応状況を比較

例) KWICの4パターンの設計比較

特性	共有データ	データ抽象化	暗黙的呼び出し	Pipe-Filter
アルゴリズムの変えやすさ	-	-	+	+
データ定義の変えやすさ	-	+	-	-
機能追加のしやすさ	+	-	+	+
処理の効率	-	-	+	+
データ定義の効率	+	+	+	-
再利用のしやすさ	-	+	-	+

対応の有無を整理するのに適する

# 設計の比較評価(2)

## ・重み付き比較テーブルで点数付け

例) KWICの4パターンの設計比較

1(低) → 5(高)

特性	優先度	共有データ	データ抽象化	暗黙的呼び出し	Pipe-Filter
アルゴリズムの変えやすさ	1	1	2	4	5
データ定義の変えやすさ	4	1	5	4	1
機能追加のしやすさ	3	4	1	3	5
処理の効率	3	4	3	3	5
データ定義の効率	3	5	5	5	1
再利用のしやすさ	5	1	4	5	4
合計 $=\Sigma(\text{優先度} \times \text{充足度})$		49	69	78	62

優先度によっての選択の違いが分かりやすい

## 設計の比較評価(3)

- その他の評価軸
  - ✓モジュール性
  - ✓テスト可能性
  - ✓セキュリティ
  - ✓利用のしやすさ
  - ✓理解のしやすさ
  - ✓統合のしやすさ

## ③費用対効果分析

- コストと効果を予測・比較する
- 例) KWICのシステム: 将来の拡張を考慮する
  - 索引が大幅に増え、処理能力向上が必要になったらどうする?
  - A) ノイズワード(a、theなど)で始まる索引の除去
    - 検索対象を削減する→フィルターを追加で対応
  - B) 同じワードを含む索引をまとめて管理する
    - 階層的に検索する→データ構造の変更で対応
  - C) 検索を行うサーバーを増やす
    - 並列に検索する→検索を割り振るコードの追加で対応

...それぞれで適するアーキテクチャが異なることが分かる



# ソフトウェアアーキテクチャ設計書

「基本設計書」ともいう

- アーキテクチャは開発全体に大きな影響を与える
  - あらゆる判断のよりどころとなる: 設計, 品質保証, プロジェクト管理
- 設計書は、設計に関する知識の重要な情報源
  - 最終的判断に限らず、検討した事項も大事

設計書に含める  
項目

## システム概要

- 主要機能や用途など、基本的事項

## ビュー

- 各ビューは、システム設計の概要を特定の視点で捉えたもの

## ソフトウェアユニット

- 開発されるソフトウェアユニットを列挙し、それらのインタフェースを記述し、ビューとの関連も示す

## 分析データと結果

- 設計の品質を評価することを目的に、設計の詳細、コンピュータリソース、実行環境を明記する(分析結果も記載する)

## 設計理念

- 設計上の選択を説明し、妥当性を記載する

## 定義、用語集、略語

- この文書を読む全ての人が同じ理解に達するために必要な事項を記載する

## ビューのマッピング

- 設計は複数のビューの集合
  - 一つのビュー(視点)では捉えきれない
- 記載するビューの数は、システム構造とその性質に依存する
  - 最低限、分解のビューと実行のビューは必要
- ビュー同士の関係も示す

# 選択基準を明記する

- 選択を行った際の判断根拠を記述する
  - 重要な問題点とトレードオフの概要を記録
- 記録する範囲
  - 判断に多くの時間を要した場合
  - 重大な判断をした場合
  - 直感的な判断と異なる場合
  - その判断を後で変更するのが難しそうな場合

# アーキテクチャのレビュー

- 工学の実践において、設計のレビューは欠かせない
- 基本設計書の品質評価
  - アーキテクチャの妥当性確認  
注文主の要求をすべて満たすことを確認する
  - アーキテクチャの検証  
奨励される設計指針に従っているか確認する

## 検証

- 推奨される設計指針にかなっていること



モジュール化、構造化され、理解しやすいか？



構造と分かり易さに改善の余地はないか？



別プラットフォームに移植可能か？



各要素は再利用可能か？



テストしやすいか？



パフォーマンスを最大限に発揮できる構成か？



誤りと障害に対応する適切な技術が埋め込まれているか？



予見されるすべての設計変更と拡張に対応できるか？

# 本日のまとめ

- アーキテクチャの評価
  - 最適なアーキテクチャを求めるための反復作業
  - 複数の案を評価し改善する
  - 定量的評価も可能
- 設計書
  - 何をどのように検討してきたかも記録する
  - たどり着いた設計の妥当性を説明する
- アーキテクチャも妥当性確認と検証をする

次回講義の事前学習:  
5.2.2, 5.2.3, 5.2.4, 5.2.5