

京都工芸繊維大学 (KIT) での 23 年

柴山 潔

2016.7.23

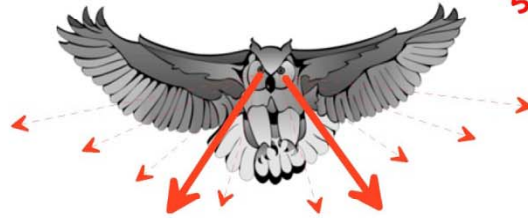
コンピュータシステム研究室(Ark)

- 1992(H4).6: 工芸学部 電子情報工学科 教授
 - 2006(H18).4: 工芸科学部 情報工学科 (改組)
 - 2016(H28).3: 退職
-
- 卒業・修了生: 229名 (うち, M修了生: 112名)
 - 留学生: 8名; 女性: 6名
 - D修了: 3名; 単位修得退学: 7名; 中退: 1名



コンピュータアーキテクトは鳥瞰する！

ちょうかん

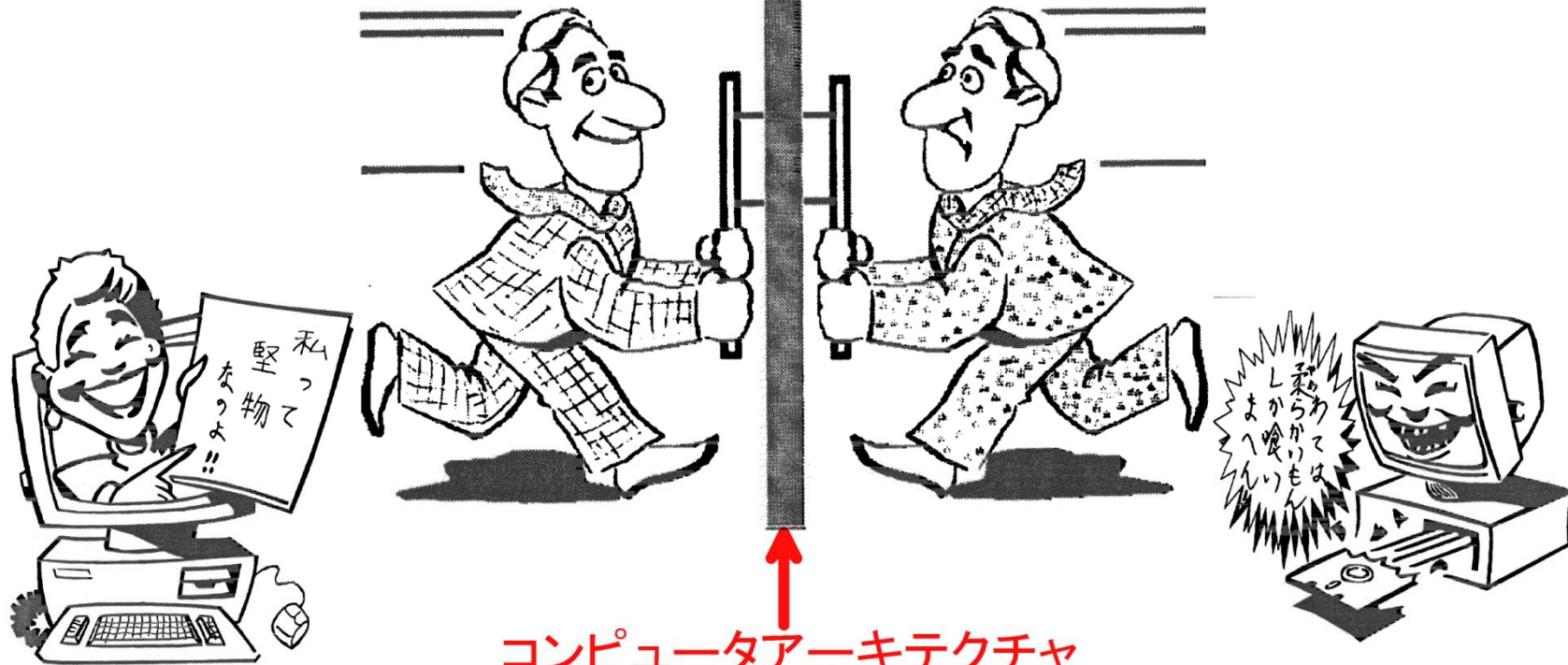


hardware

ハードウェアの間

software

ソフトウェアの間



コンピュータアーキテクチャ
computer architecture

鳥瞰(ちょうかん)(1)

- 大局観, グローバル的あるいは巨視的観点での考察が鳥瞰すること
- 自分が歩いている道と立ち位置を鳥瞰すれば, 見えない自分の足跡や行く先また周囲の状況や情勢が見える
- 鳥瞰はグローバルな評価

鳥瞰(ちょうかん)(2)

- 自分の立ち位置を鳥瞰するには、自分を俯瞰する位置に眼を置かねばならない、それは一種の客観的評価につながる
- 鳥瞰はマクロでグローバルな把握になくてはならない
- 行き詰まったら鳥瞰を
- 時々、自分の立ち位置を鳥瞰しよう

グローバル(global)化(1)

- **グローバル化**とは、個（代表例：自分，個人）を全体（代表例：世界，地球）に置いて，その全体の中での個の立ち位置を認知すること
- **グローバル化**は鳥瞰によって得ることができる
- **グローバル化**が対象とする全体は，世界や地球などの空間的全体（空間**グローバル化**）とは限らない．むしろ，現在・過去・未来と連なる時の流れすなわち時間的全体（時間**グローバル化**）も

グローバル(global)化(2)

- 個人(自分)における時間**グローバル化**とは、一生(生涯, 人生)の時々、自分の過去から未来を**俯瞰**して自分の現在を見つめ直すこと
- 時間**グローバル化**での見つめ直しに空間**グローバル化**を併用すると、**グローバル化**が増進

グローバル(global)化(3)

- ローカル(local)の特色や特徴を保護さらには改革せずに平板にグローバル化すると、ローカル色は薄まる
- ある量のものをグローバル化すると、そのものの量だけでなく質も薄まるから、グローバル化では、量だけを増やすだけでなく、質の保持さらには向上が大切

PDCA(Plan-Do-Check-Act)サイクル(1)

- PDCAサイクルはPlan(設計, (広義の)デザイン, 企画)ステージで始まる, また始める
- Do(制作, 製作, 作成)の前にはPlanステージが, Doの後にはCheck(評価)ステージがある
- Act(改善, 改良)ステージはフィードバックでPDCAサイクルには必須

PDCA(Plan-Do-Check-Act)サイクル(2)

- ものづくりはPDCAサイクルそのもの
- Plan(設計, デザイン)ステージにおいて列挙する選択肢をCheck(評価)することも
- Check(評価)は, Do(制作, 製作, 作成)に対してだけでなく, Plan(設計, デザイン)に対しても行うべき

PDCA(Plan-Do-Check-Act)サイクル(3)

- 2次元での単なるサイクル(繰り返し)ではなく、3次元のスパイラルアップ(上昇らせん)PDCAサイクルを目指そう
- ときには一服してPDCAサイクルを鳥瞰してみよう
- PDCAサイクルは鳥瞰せねば見えない
- PDCAサイクル全体を総合的に眺めることが「鳥瞰」に通じる

方針と機構の分離(1)

- 設計(デザイン)では必須
- ものづくりには**方針**(**ポリシ**; **policy**)も**機構**(**メカニズム**; **mechanism**)もどちらも必要
- ものづくりでは, **方針**(**ポリシ**)はつくるものの心(精神, 理念, 思想), **機構**(**メカニズム**)はつくるものの体(肉体, 実際, 実体)

方針と機構の分離(2)

- ひとつの**方針**(**ポリシー**)で、複数の**機構**(**メカニズム**)を設計して、それらを選択肢として列挙しよう
- 設計(**デザイン**) = **方針**(**ポリシー**)の確立
- 設計**方針**(**デザインポリシー**) = 設計思想(**デザインフィロソフィ**)
- シンプルだけれど骨太な**設計方針**(**ポリシー**)が強靱でしなやかな**機構**(**メカニズム**)を生む

ホワットツ－(what to)(1)

- 技術のホワットツ－(what-to)は「その技術によって何を実現したいのか、あるいは、何を実現するのか?」、「その技術によって何をしようとしているのか?」、「その技術の目標は何か?」、「その技術は何の役に立つのか?」
- ハウツ－の追究にホワットツ－は必須
- ホワットツ－の探究を開発の動機にしよう
- ホワットツ－があれば開発は楽しい

ホワットツツー(what to)(2)

- **ホワットツツー**は技術開発の肝
- 骨太で堅固な**ホワットツツー**の裏打ち(裏付け)のある**ハウツツー**(技術)は強靱でかつしなやか
- ひとつの**ホワットツツー**にいくつもの**ハウツツー**が選択肢としてぶら下がる
- **ホワットツツー**は理念, **ハウツツー**は実際, ものづくりではどちらも必須

デザイン(広義)

- 広義の「デザイン(する)」は「設計(する)」、「立案(する)」、「企画(する)」、「構想(する)」の意
 - 広義の「デザイン(する)」は「意匠(を作る)」、「図案(を描く, かく)」、「下絵(をかく)」の意である狭義の「デザイン(する)」とは異なる
- 工学(ものづくり)における設計は(広義の)デザイン
- 広義の「デザイン(する)」は、触媒となって、科学と工学と技術とを、合成する、統合する、融合する、混合する、組み合わせる、連携させる、結び合わせる、くっつける

設計(plan, design)

- **設計**は企画や(広義の)デザインと同じ
- **設計**力とは, 構想力, 問題設定力, 創造力, 表現力
- **設計**(する)とは, 構想を練りその構想を表現する, 問題を設定する, 創造する, 継続的に計画及び実施する, 制約条件下で解を見出す, (工)学と技(術)を総合的に応用する, こと
- ものづくりは**設計**で始める
- **設計**がエンジニアリングデザインの肝

設計思想(design philosophy)(1)

- 設計の方針(ポリシー)が設計思想(デザインフィロソフィ; design philosophy)
- ものづくりでの設計とは「ひとつのポリシーを見つける」こと
- ひとつのポリシーに基づく複数メカニズムの設計(=ポリシーとメカニズムの分離)を目指そう

設計思想(design philosophy)(2)

- ものづくりのPDCAサイクルでの最初のPlanステージが「方針(ポリシ)の設計」, すなわち, 「設計思想の確立」ステージ
- ものづくりにおける設計思想に基づく方針(ポリシ)とは, ものづくりにおけるホワットツ-

エンジニアリングデザイン(1)

- エンジニアリングデザイン(工学設計, 設計工学; engineering design)のデザイン(設計)は広義
- 工学(engineering)と(広義の)デザインとの融合
- 工学を(広義の)デザインする
- 工学での学修に(広義の)デザイン手法を利活用する

エンジニアリングデザイン(2)

- 必ずしも解がひとつでない課題に対して、実現可能な(複数の)解を見つけ出して比較できる
- エンジニアリングデザイン力の要は設計力
- エンジニアリングデザイン力は、ものづくりのPDCAサイクル全般を総合的に見渡せて、かつPDCAのどのステージ(過程)もこなせる能力

開発(1)

- **開発** = 工学 × 技術, 工学(engineering)の裏付けがある技術(technology), 技術の裏付けがある工学
- **開発** = ホワットツ－ × ハウツ－, **開発**には必ずホワットツ－がある
- **開発** = 方針(ポリシ) × 機構(メカニズム), **開発**には方針(ポリシ)が必須

開発(2)

- **開発**も設計(デザイン)で始める
- **開発** = エンジニアリングデザイン, **開発**に必要なのはエンジニアリングデザイン力
- **開発**者 = 高度専門技術者(高度専門職業人)

広報と宣伝(1)

- 「技術の**宣伝**」とは「**設計思想の広報**」
- 技術論文には「**設計思想の明示**」が必須
- 設計思想**を「**ホワットツ**」で具体的に語れば、技術論文ができる
- 「ものづくり」とは「(つくるものの)**設計思想の具象化**」

広報と宣伝(2)

- 「もの」や「技術」の売り込みとは、設計思想のアップール、**広報**、**宣伝**、啓蒙
- 技術の**広報**や**宣伝**は技術者の役割
- 教科書にも執筆思想が要るが、それはその教科書で述べる技術の設計思想

トレードオフ(tradeoff)(1)

- **トレードオフ**(tradeoff)は「同時には成立しない**二律背反**の関係」, すなわち, 「あっち(あちらを)立てればこっち(こちらが)立たずの関係」
- **トレードオフ**の**設計**とは**トレードオフ**の調停で, 最適な**トレードオフ**(状態)を見つけること
- **トレードオフ**は**鳥瞰**せねば見えない

トレードオフ(tradeoff)(2)

- 自然界のトレードオフは自然に調停され，人工物のトレードオフは人工的に調停される
- ものづくりは(ものの)目的の達成を目指す最適トレードオフの設計
- コンピュータシステムのトレードオフはハードウェアとソフトウェアの機能分担，すなわちコンピュータアーキテクチャ

高度専門技術者(高度専門職業人)(1)

- 複数のメカニズムをひとつのポリシで設計する技能を持っている人
- ひとつのポリシに基づく複数のメカニズムを選択肢として列挙し, それらを比較し評価できる技術者(職業人)
- 工学を設計できるエンジニアリングデザイナー
- ハウツーだけではなく, ホワットツを語る技能を保持

高度専門技術者(高度専門職業人)(2)

- ものの設計((広義の)デザイン)ステージにおいて、作るもののホワットツ-を明示できる技能を修得
- ものづくりのPDCAサイクル全般を総合的に見渡せて、かつPDCAのどのステージ(過程)もこなせるエンジニアリングデザイン(能)力を持っている、ものづくり学の実践者(ものづくり技術者)

高度専門技術者(高度専門職業人)(3)

- エンジニアリングデザイン(能)力は必須
- エンジニアリングデザイン力を身に付ければ,
(専門技術者は)高度専門技術者に成れる

教科書執筆でのポリシー(執筆思想)(1)

- **教科書**は一定の範囲の学問対象をカバーする**原理**を分かり易く紹介する道具
- 著者以外にその**教科書**を使う(採用する)教員が**多数**存在することが良い**教科書**
- **教科書**は学習者(学生)だけではなく**教員**もオルグするために執筆すべき

教科書執筆でのポリシ(執筆思想)(2)

- **教科書**はその使用者である教員と学習者(学生)の両方に明瞭で確固たる指針を示す**羅針盤**(ガイドブック)
- **教科書**は, その対象分野に対する執筆者(著者)の**オリジナルな設計思想**に基づいて展開すべき
- 1冊の**教科書**で1つのテーマが簡潔で完備しており, また, そのテーマである**原理**が小説のように読者の心に残るべき

「ハードウェア入門」

- コンピュータ(システム)についての知識が無い人向け
- コンピュータ → ハードウェアとソフトウェア → 内部装置と外部装置 → プロセッサとメインメモリ → 演算と順序制御 → 論理回路と論理素子 というように、コンピュータ(システム)について、シェルからコアへ、トップダウンに紹介
- コンピュータとコンピュータシステムとの違いも明確化



「コンピュータアーキテクチャ」

- 「コンピュータアーキテクチャの定義 = ハードウェアとソフトウェアとの機能分担」を実際に展開
- 方式説明のすべてに、「ハードウェアとソフトウェアとの機能分担方式」を指標とする長所と短所とを個条書きで明記
- 「システムプログラム（コンパイラ，OS）を支援するアーキテクチャ」の章を設ける



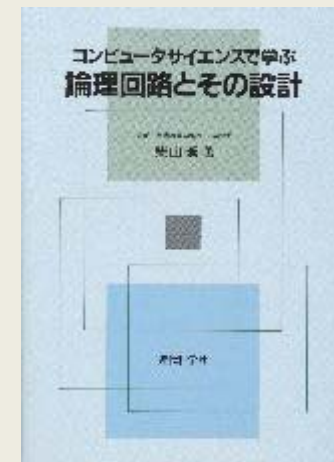
「コンピュータ工学への招待」

- 情報系学問や技術（ICT）の基盤（インフラ）である**コンピュータ**に関する**工学・技術分野**を1つのコンパクトな学問分野として整理・統合
- **コンピュータシステム**を構成するための基本ハードウェア機構である**論理回路**，基本ソフトウェア機能である**オペレーティングシステム**，それらハードウェアとソフトウェアの機能分担方式として定義できる**コンピュータアーキテクチャ**の代表例について考察



「コンピュータサイエンスで学ぶ 論理回路とその設計」

- コンピュータシステムを構成する基本ハードウェア機構である論理回路とその設計原理を(論理)数学だけをベースにして説明
- 特に, コンピュータにおける(離散)時間やメモリの概念を(論理)数学だけで扱えることを組み合わせ回路の発展形である順序回路を題材にして紹介



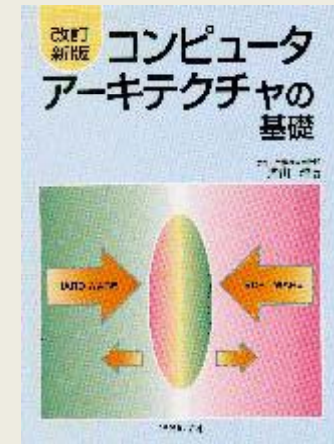
「コンピュータサイエンスで学ぶ オペレーティングシステム —OS学—」

- 固有名詞のオペレーティングシステム全部を貫くOS原理について固有名詞を使用せずに整理して体系的に明示
- OSは、ハードウェアと(ユーザ)プログラムとのインタフェースとなるソフトウェアであり、それぞれから互いを隠ぺい(仮想化)し、かつ相互の通信(同期)を一元化して代行
- OS原理の基盤は「割り込み」



「コンピュータアーキテクチャの基礎」

- 「コンピュータアーキテクチャは、コンピュータシステムにおけるハードウェアとソフトウェアの機能分担方式（トレードオフ）」と定義
- 「コンピュータアーキテクチャの設計」は、ハードウェアとソフトウェアとの機能分担方式を設計（トレードオフを調整）すること
- 機能分担を変えれば、（別の）方式ができる



原理(公理)

- (ノイマン型)コンピュータの原理は「論理数学(2進数表現)とソフトウェアの存在」
- ハードウェアの基礎である論理回路は情報工学の基盤である「(情報)数学」だけで語れる
- オペレーティングシステムはハードウェアと(ユーザ)プログラムとを仲介する基盤ソフトウェアであり、それぞれを他から隠ぺい
- コンピュータアーキテクチャは「コンピュータシステムにおけるハードウェアとソフトウェアの機能分担方式(トレードオフ)」

コンピュータアーキテクトは鳥瞰する！

ちょうかん

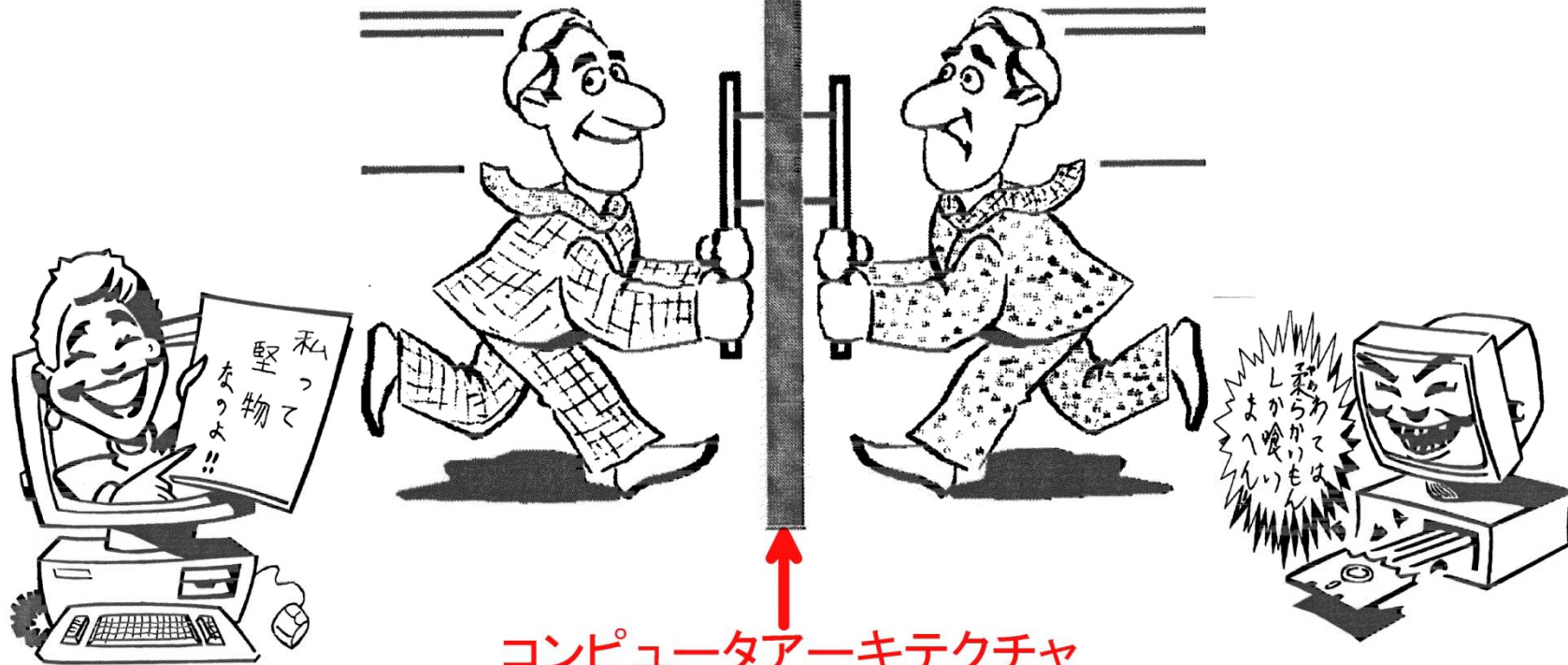


hardware

ハードウェアの間

software

ソフトウェアの間



コンピュータアーキテクチャ
computer architecture

