

## コンピュータ概論B

－ ソフトウェアを中心に －

### #07 アプリケーションソフトウェア

京都産業大学  
安田豊

## アプリケーションソフトウェアとは

- 教科書 pp.96-
- その昔
  - 解きたい問題はすべてプログラムを書き起こして処理した
  - 100の問題に100のプログラミング
- 現在
  - 「良くやる処理」のプログラムはツールとして一般化
  - これが今日のアプリケーションソフトウェア
  - ツール（道具）の選択はユーザの判断
- 本講義では
  - ツールの特性、処理の考え方を説明
  - 使い方などは説明しない

## 典型的アプリケーション

- 年賀状宛名印刷ソフト
  - 特定用途に特化されている
  - データ入力、管理、レイアウト、印刷、という必要な手順が全て込み
  - 手順に従って操作すれば成果が得られる
  - 楽が出来る
  - コンピュータの理解が軽くて済む
- これがアプリケーション
  - 「応用ソフトウェア」

## 利点

- ユーザは問題解決に集中できる
  - 文書を作っているときに、印刷の手順など考えたくない
  - 文字の並びだけ、体裁だけ考えたい
  - 統計処理を行っているとき数値計算法を考えたくない
  - データの流れ、関係だけに注目したい
- 工数を減らせる
  - やり方が確定しているものを自分でまた書きたくない
  - 誰かが書いてくれたら使うだけで手間が済む
- 処理結果がきれい
  - 専門のプログラマが手間暇掛けて作っている消費などは見た目がきれい
- ゴール：簡単な操作でより良い結果を出したい

## ブラックボックス

- 操作の簡略化＝処理の隠蔽
  - すべてを指示しなくても機械が勝手にやる
  - 問題解決法がブラックボックス化されている
  - アクセルを踏めば、前に進む
- 問題点
  - 処理結果の解釈を誤ったりする
  - この計算の精度はどのくらいか？
  - 平均を取って見たが値なしのデータは数に入っているか？
- そういった問題点が無い用途に最適
  - ワープロ作業など「できておればよい」ようなもの

## ブラックボックス

- 周辺の問題点
  - マニュアルの不備がありがち
    - 不親切、わかりにくい
  - 処理結果が異常になったときに対応できない
    - 何が原因でこうなったのかが特定できない
  - 細かいところでやりたいことが出来ない場合がある（出来上がりが画一化する）
- それでも
  - プログラムの再利用は重要
  - アプリケーションの発達・普及は重要

## 対処法

- 教科書 pp.99-
  - しかし少し無茶
- せいぜい
  - 結果の検証手段をもっておく
  - 中でどのような処理が行われているのか意識する
  - バグから身を守るためにユーザ登録して修正版を使う

## 各アプリケーションの紹介

- ツールの特性を知る
  - よりよい道具の選択をする
  - アプリケーションの潜在的問題を知っておく

## かな漢字変換ソフト

- 教科書 pp.99-
  - タイプライタ・キーボードで数千字の漢字を入力する
  - かなを漢字に変換する方法が主流になった
  - もっとも身近な人工知能的なソフト
    - 意味解析より大量辞書
  - 文字コードの不足
    - JIS 数千字では全く足りない
    - 文化的活動の制約にならないように

## エディタ

- 教科書 pp.100-
- もともとはプログラムエディタ
  - 電子メールの本文、簡単な文書作成などに使われている
  - ワードプロセッサが「重い」ため
  - テキストファイルを専門に扱う  
(ワードプロセッサはテキストファイルではない)

## ワードプロセッサ

- 教科書 pp.100-
  - テキストエディット機能に加えて
  - 装飾、体裁加工がついている
  - 推敲、校正などさまざまな補助機能
  - きれいな印刷で出力する
  - これらの補助情報のためにデータファイルは専用フォーマットとなっている
- 清書機（例えばタイプライタ）ではない
  - 再利用、推敲が可能
  - これに向かって文章を書くというスタイルを実現
  - 専用機もある（減ってはいるが）

## DTP ソフトウェア

- 教科書 pp.102-
- Desk Top Publishing
  - 自由度の高い組み版が扱える
  - 画像
  - PostScript など高品位な印刷に耐える精度
  - 出版社へはDTPソフトの出力データで入稿するスタイルが普通に
  - フォントの整合性など職人芸も多く含まれる
  - 道具として良いところも悪いところも残す好例

## 表計算ソフト

- 教科書 pp.102-
- 表形式 + Cell による自動計算
  - Spread Sheet
  - アイディアは VisiCalc (1977?) から
  - Multiplan, Lotus 1-2-3, Excel ...
  - ビジネスに大きくパソコンを食い込ませた立て役者
- 現在でも多用
  - ビジネス、簡単な統計分析など

## 表計算ソフト

- きわめてアプリケーション的
  - 「VisiCalc が欲しいんだ」と PC Shop で叫ぶ客
- まさにブラックボックス
  - 意味が分からなくてもグラフは描ける
  - 特定値があっても平均は出る
- 結果の意味を吟味するには
  - データの中身とソフトの振る舞いを知り
  - 自分がやろうとしている仕事の中身を理解する
- Garbage In, Garbage Out

## グラフィックソフト



- 教科書 pp.105-
  - 読みは「グラフィクス」が正解?
- 画像を扱うため
  - ビットマップ形式とベクトル形式の使い分け
- 色の表現
  - 8bit color : 256色 / 12bit color : 4096色
  - 24bit color : 約 1600 万色 (16,777,216) といっても RGB 256 段階で決して十分ではない
- 3D / 動画への展開
  - 今後は静止二次元画像だけではない
  - まさにコンピュータでしかできない分野



## CAD / CAM / CAE

- 工業向けのコンピュータ支援ソフト
  - Computer Aided Design - 設計支援 (製図)
  - 図面の再利用が簡単
  - 3D モデリングなども可能に
  - Computer Aided Manufacturing - 製造支援
  - CAD設計したデータをCAMに渡し、効率よい業務の流れを作る
  - Computer Aided Engineering - 研究開発支援
  - 開発環境全体を支援 (定義は曖昧のよう)
- 80年代から90年代の Unix EWS の市場

## 提案：柔軟なアプリケーション

- 二者択一の現状モデル
  - 低レベルのプログラミングか、
  - ブラックボックスとしてのアプリか
  - 開かれたアプリケーションはないのか?
- 考え方
  - ブラックボックスではない
  - 再利用が容易なもの
- Squeak (Smalltalk)
  - 2003 年度京都で