情報処理の概念

#2 デジタル化、汎用性、フォーマット

Yutaka Yasuda

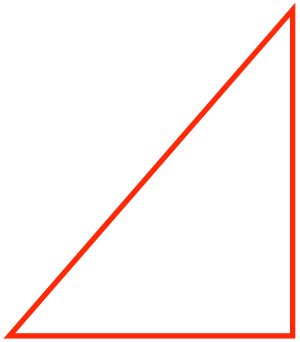
デジタルで表現する

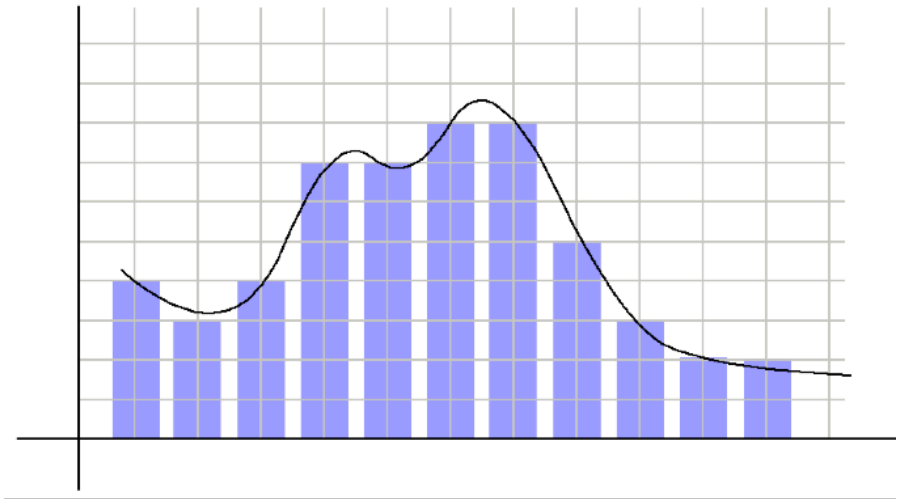
- すべての情報を数値（符号）で表現すること
  - その方法
  - その価値

  について、具体的な例を示しながら説明する

デジタル処理

- 情報のデジタル化＝符号化＝数値化
  - 三角形なら
    (0,0),(100,0),(100,210)
  - 音声なら
    4,3,4,7,7,8,8,5,3,2,2…
- 確定的な数値として表現
- 欠点と利点の双方をもつ



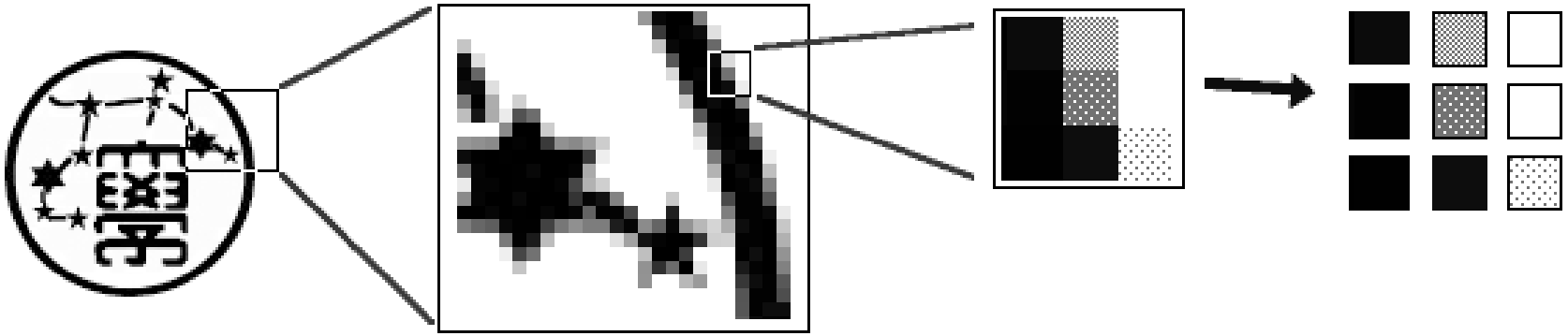


デジタルデータの特徴

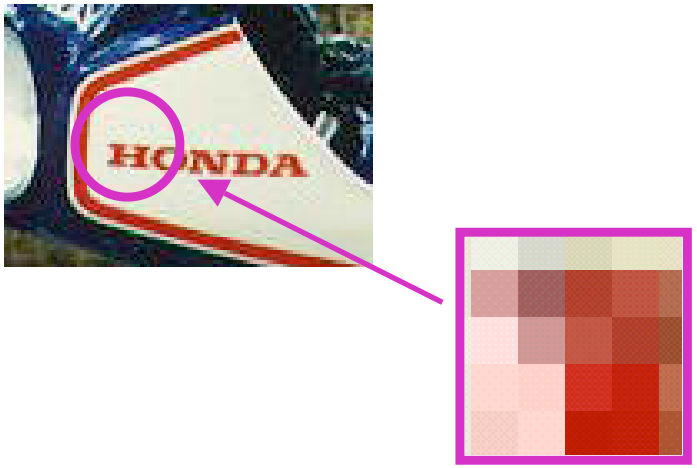
- 完全な複製
  - 複製・通信・保存に伴うノイズの除去
  - 完全さの検証が可能に
- 不完全なデータ化
  - 初期ノイズの発生（近似でしかない）
- 考え方
  - 初めに精度を決めることでそれ以後の精度以内の変化をゼロにした
- 利益
  - 数学的なテクニックが多く適用可能に
  - コンピュータによる知的な自動処理が可能に

画像のデジタル表現

- 絵は画素(Pixel : Picture Element)ごとに分解
  - 各画素ごとに数値化
  - 1-3-5,1-2-5,1-1-4 のごとし
  - 空間的なサンプリングと考える



カラー画像のデジタル表現（例）



一画素ごとに赤・緑・青 (RGB) に
色分解して各色256段階で記録

	赤	緑	青
→	230	29	10
→	180	28	9
→	230	22	17

空間的サンプリングと考えれば良い

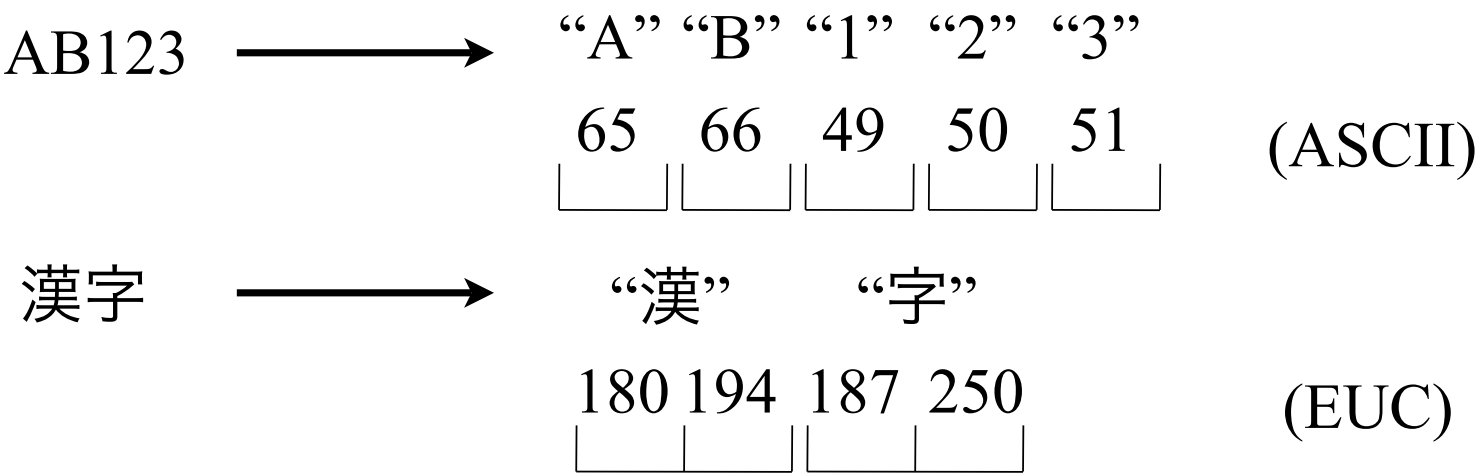
動画も簡単にデータ化できますね？

文字のデジタル表現

- 数値化された文字、とは？
  - あり得る文字にすべて番号を振る
  - 番号付け＝コード化（符号化）
- 元もと文字はデジタルな存在？

  例：アルファベットと数字からなる情報を符号化する
  - ABC = 1,2,3 とすれば 26 で足りる
  - abc = 27,28,29.. で 52 まで
  - 0,1,2 = 53,54 で 62 まで
- 漢字はたいへんだが 6 万もあれば？

文字のデジタル表現（例）



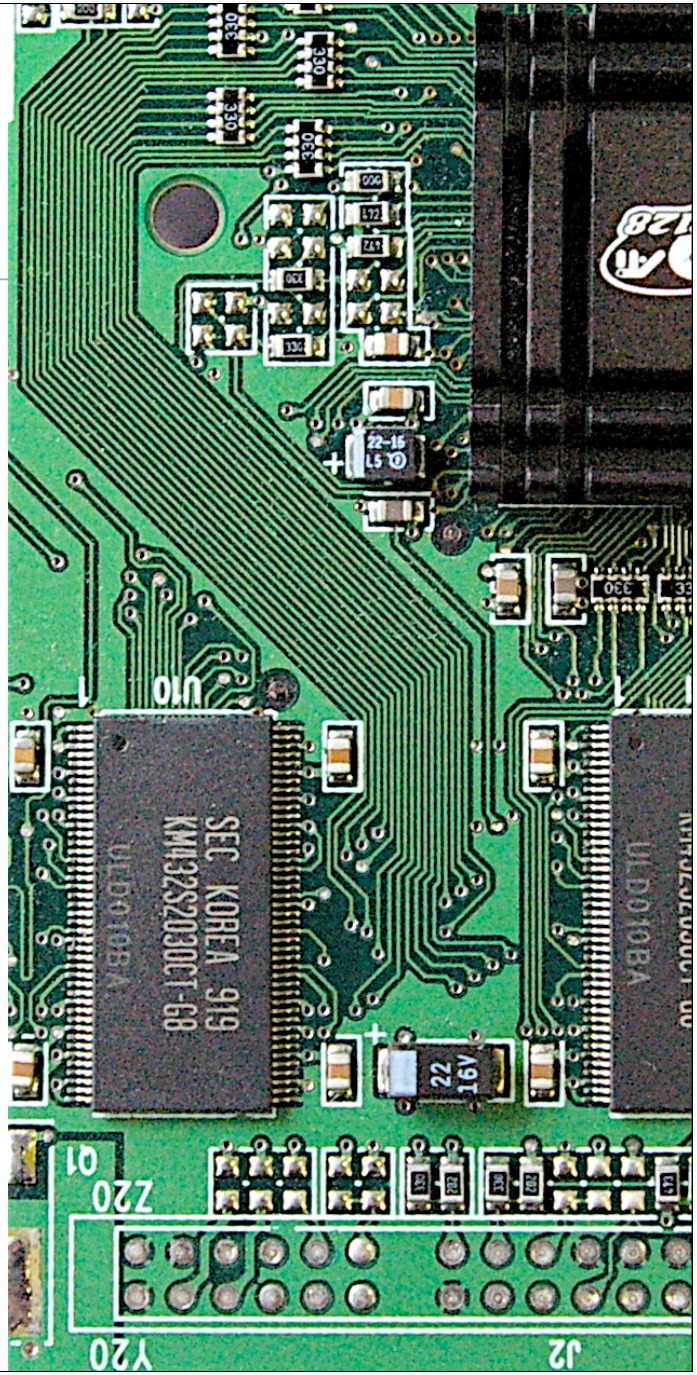
データをおさめるための「枠」があると便利

Byte : データの標準枠

- Byte （バイト）
  - データをおさめる（値を入れる）ための枠
  - 一つの枠には0-255までの256種類の値が入る (8bit)
  - 255を超える値は二桁（2Bytes）使う
  - Byte is not ‘Bite’ , bit is not bit
- ASCIIは 1 バイト
- 漢字は（普通は） 2 バイト
  - 「フロッピー1枚は新聞何枚に相当し、、」

bit：データの最小単位

- コンピュータ内部は電気配線
- 配線に電気が通っている、いない、だけで処理（符号は2種だけ）
- 「0と1（二進数）で動作」の実体
- 配線 1 本が 1 bit に対応
- データ種類ごとに再配線できない
- 一定本数によるデータ表現＝「枠」



Byte量：音楽CDは何バイトあるか？

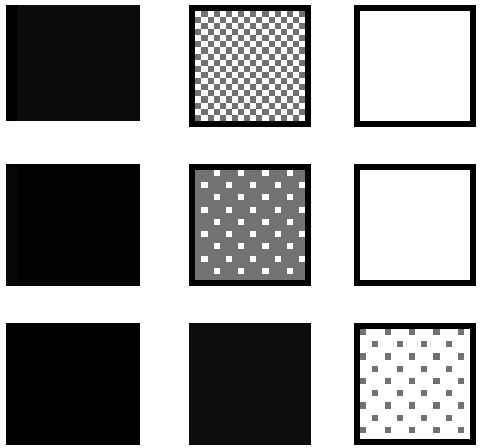
- さまざまなもののバイト数
- 広辞苑 (第二版)
  - 24字 x 50行 x 4段 x 2400ページ=11,520,000 字
  - 一文字 2 Bytesとして 23 Mega Bytes
- 音楽CD
  - 44KHz x 65536段階(2Bytes) x 2ch = 176KB/sec
  - 176KB x 3600sec = 633,600 KB = 634MB
  - さまざまなものが bit にかわる姿を想像できたろうか？

デジタルデータ

- その実体は数値（記号）の列
  - 音声：111,121,122,89,80,82,75….
  - 静止画：10,240,22,30,34,80…
  - 音声付き動画：12,33,45,1123,488…
  - 文字：33,38,42,60,32,39,55,80…
- これだけでは利用できない（意味が汲み取れない）
  - 符号化ルールとデータは常に一体
- このルールがフォーマット（書式）を生む

フォーマット（書式）

- 同じ画像データでも数え方を違えると全く違う数字列に
  - 1-3-5,1-2-5,1-1-4
  - 1-1-1,3-2-1,5-5,4



- 符号化ルールと一致する復号化をしないと異なる結果に

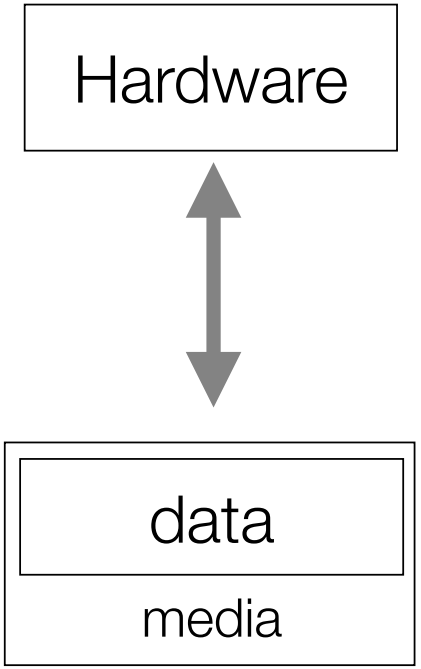
フォーマット（書式）

- デジタルデータを解釈するには
  - 解釈（復号）ルールが必要
- フォーマット（書式）
  - つまりデータにはフォーマットがある
  - フォーマットを間違えて解釈すると間違った結果が導き出される
  - 異なるアプリケーションでデータが扱えない理由
- （データにおける）「互換性」という概念の実体

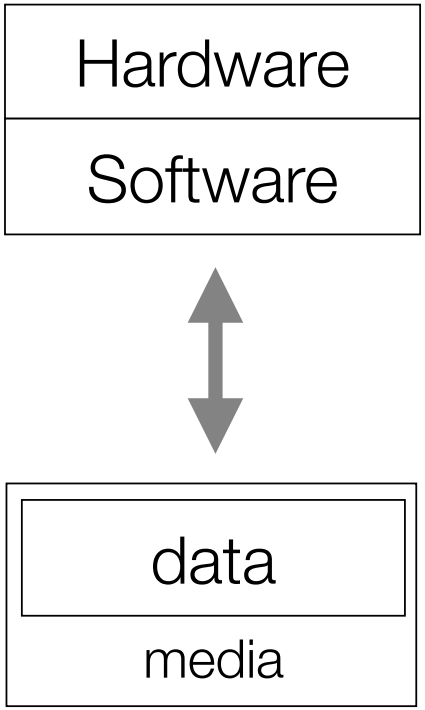
文字におけるフォーマットの問題

- 統一されていないルール
  - 文字番号表（この字を何番とするか）はいくつかある
  - バイト単位での並べ方（次の 1 バイトは漢字の前半か、後半か）にも幾つか
- アルファベット：ASCII コード
- 漢字（日本語）：JIS/EUC/Shift-JIS漢字コード
- Unicode
- いわゆる文字化けの原因

アナログシステムとデジタルシステム

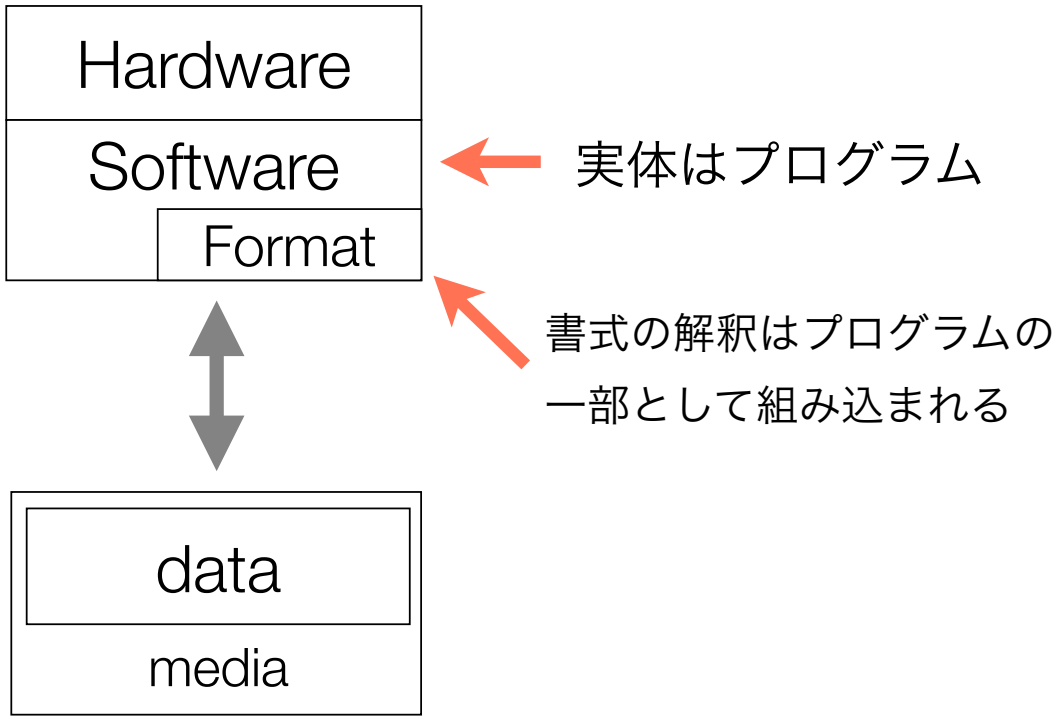


典型的なアナログシステム
（レコードプレーヤーなど）



典型的なデジタルシステム
（コンピュータなど）

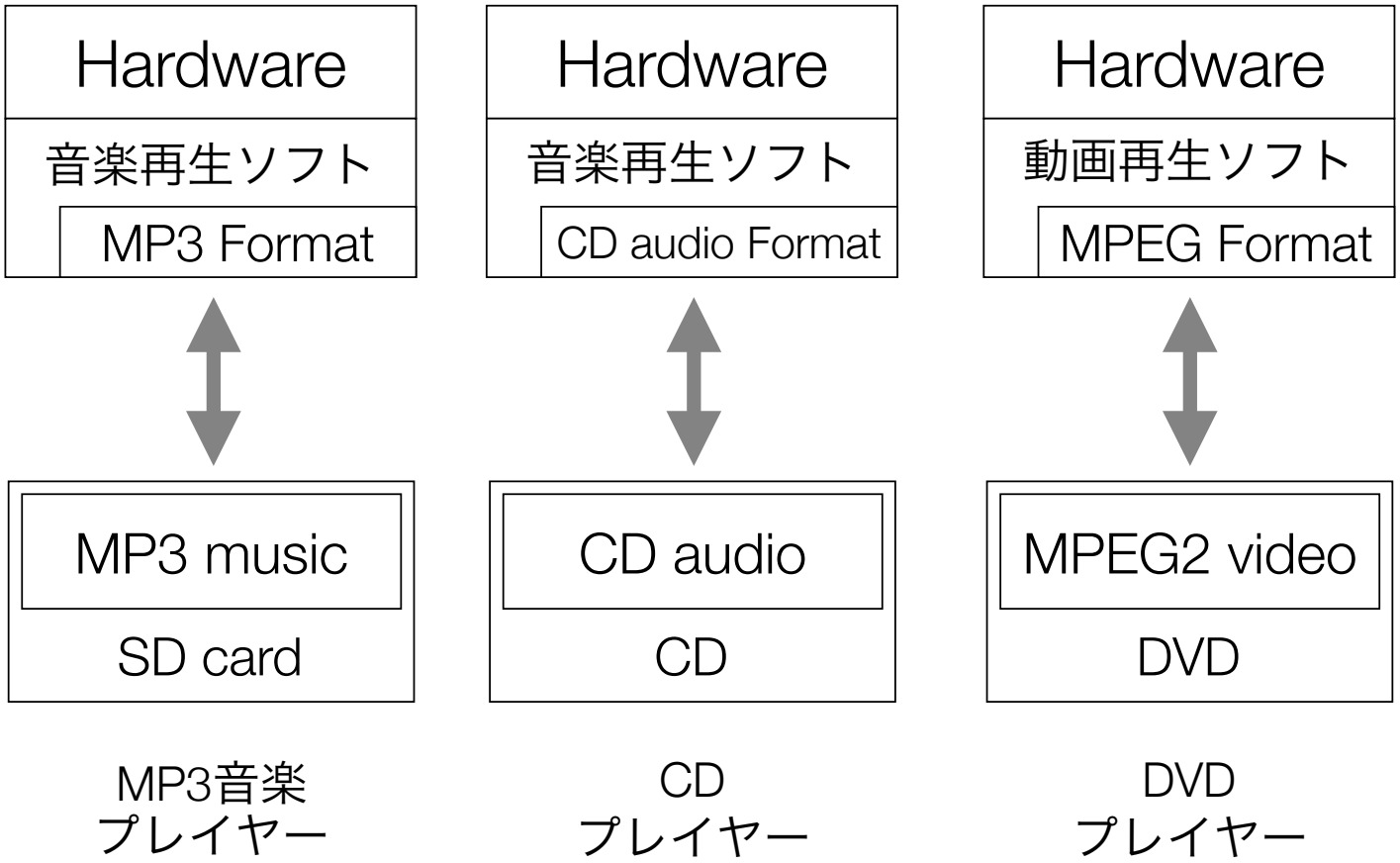
書式とデータの関係



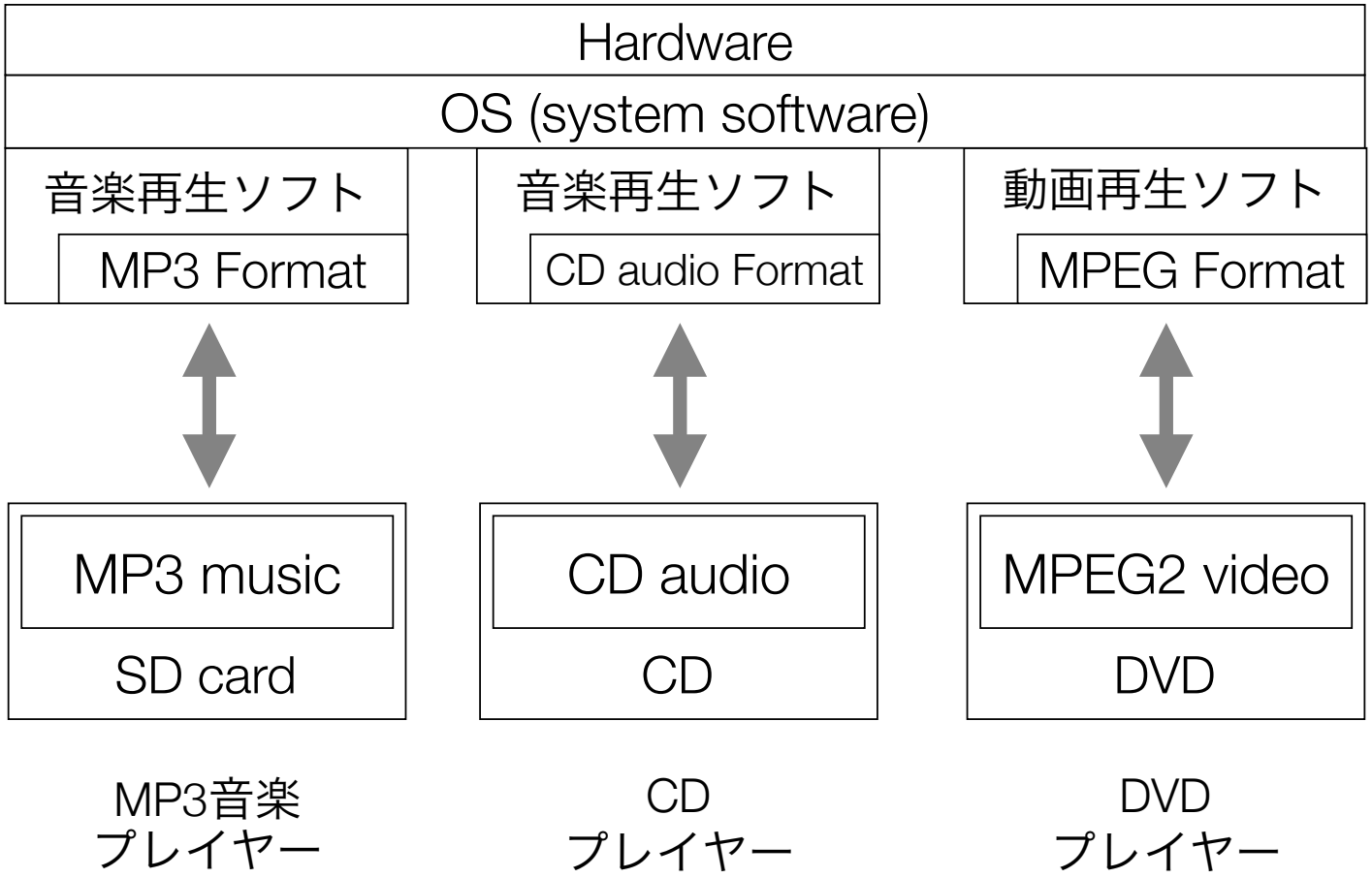
典型的なデジタルシステム

（コンピュータなど）

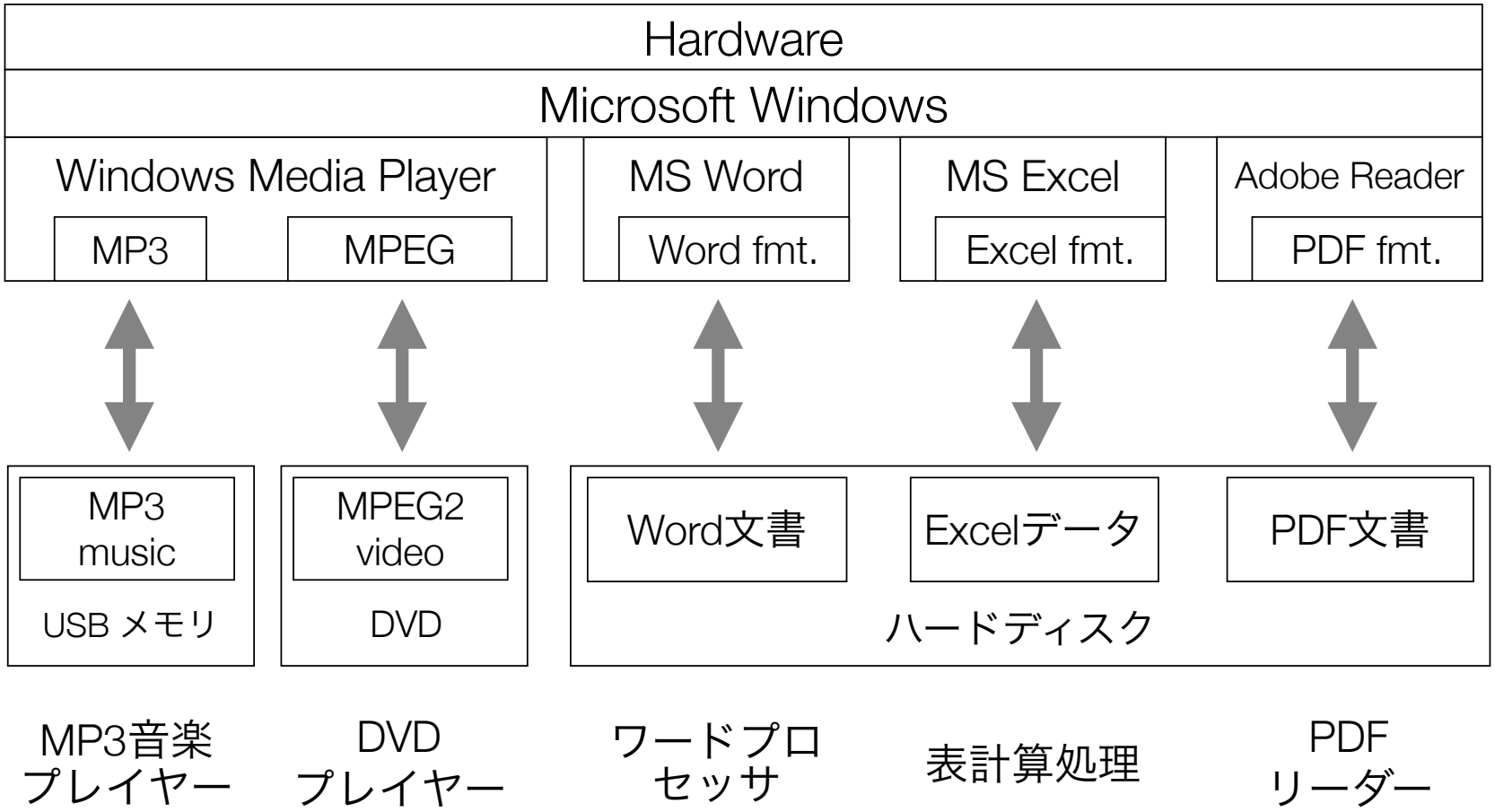
デジタルシステムの柔軟性



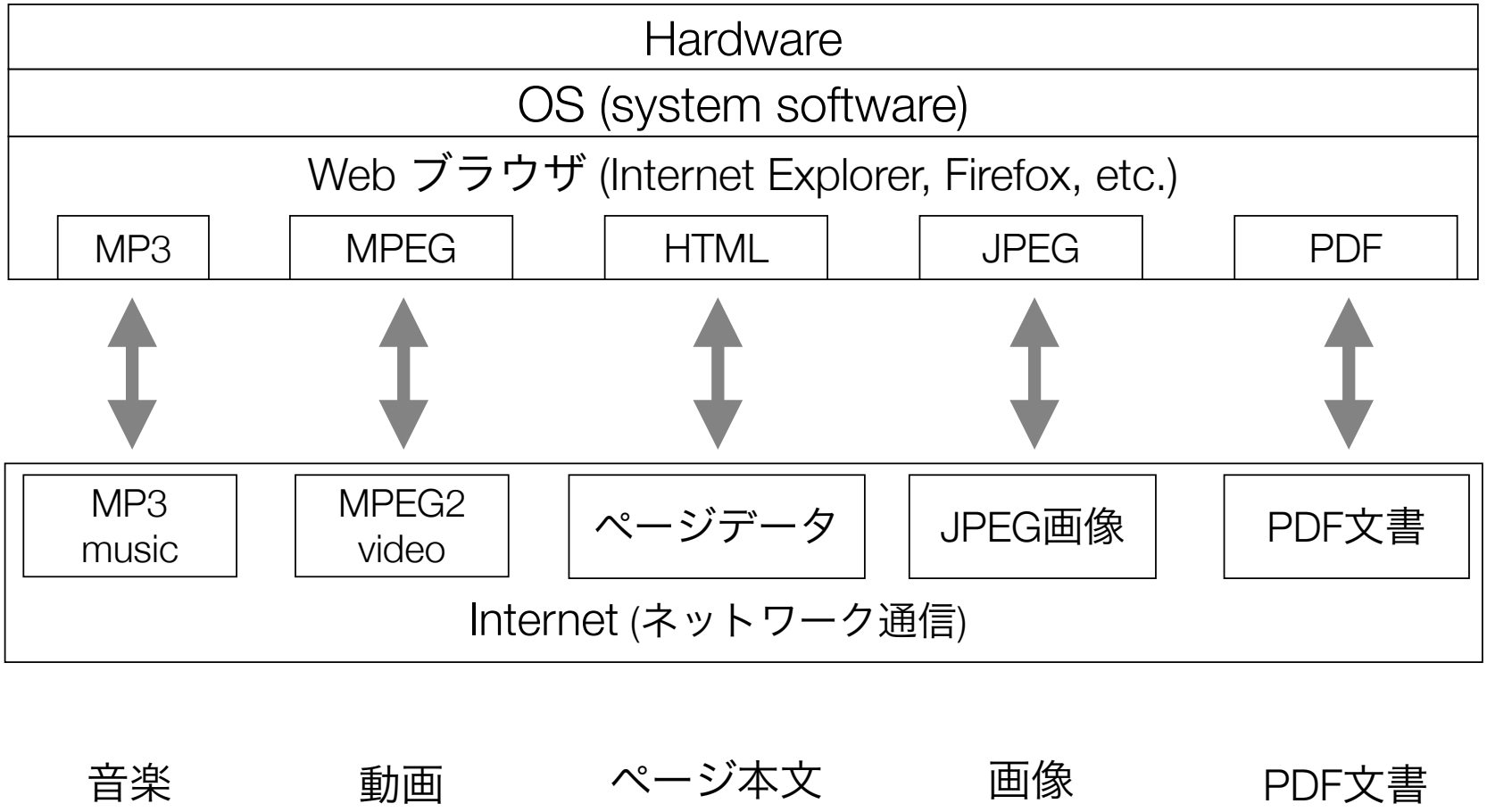
PC : 汎用デジタル処理システム



いつも使っている Windows パソコン



Web ページ閲覧におけるデータ処理



デジタル化のインパクト

- 汎用性
  - 情報はフォーマットと値で表現される
- 汎用(generic)のものに特定(specific)の機能を載せる
  - 汎用データ通信網に特定用途サービスを載せる
  - このサービスを汎用コンピュータに特定用途アプリケーション・ソフトウェアを載せて実現
  - ソフトウェアを入れ替えて新しい機能を実現可能
  - ソフトウェアで対応することの柔軟性

まとめ

- デジタルデータのメリット
  - 完全な複製
  - デジタルコンピュータによる自動処理
- デジタルデータとフォーマットの関係
  - デジタル化で情報はメディア（物理的制約）からは自由になったがフォーマットが重要になった
  - 互換性という概念
- デジタル化のインパクト
  - ソフトウェアによる柔軟性

事例紹介

- Microsoft の HD DVD への進出
  - 動画フォーマットとしての Windows Media Series 9 の提出が意味するものは何か？
  - NEC / 東芝は MPEG など公開の場で作られたフォーマットを推している
  - 何故か？