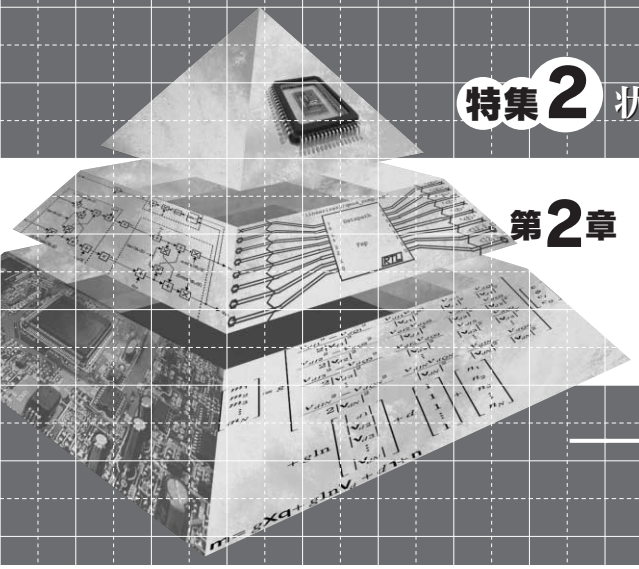


第2章 ソフトウェアはこうモデリングする！

――階層的に抽象化して利用しやすいモデルを作る

山崎辰雄

本稿では，ソフトウェアのモデリング例として，カード型携帯端末「アカウント・バンク」のデータ格納部分であるシリアルEEPROM（electronically erasable and programmable read only memory）をモデリングする．状態遷移モデル（状態遷移図，状態遷移表）を作成して，実装（アルゴリズム）の検討まで行う．（編集部）

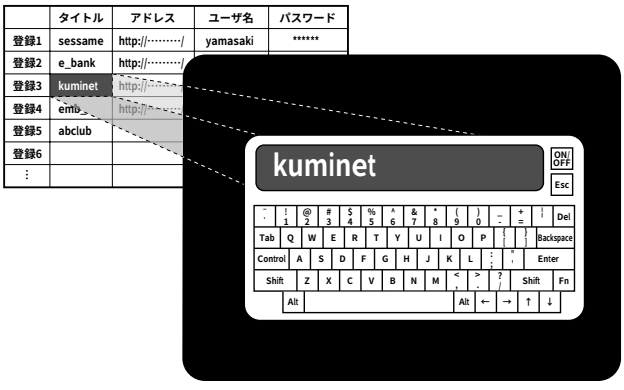
ソフトウェアのモデリングの手順や考えかたそのものは，システムのモデリングと同じです．本稿では，モデリングの成果物（モデル）をアルゴリズムのレベルで実装するところまでを示します．

1.データ格納の検討とモデル概要

はじめに，カード型携帯端末「アカウント・バンク」のシステムの中核となるデータ格納について検討し，シリアルEEPROM（electronically erasable and programmable read only memory）がシステム内でどういう位置づけにあるかを確認します．

アカウント・バンクは電池が切れたときも登録データを保持しなければならないので，不揮発性メモリを用いることにします．ここでは，接続端子数が少なくてパッケージ・サイズの小さいシリアルEEPROMに格納します．

アカウント・バンクの表示画面は，登録されているデータに対して一度に一つの登録データ，それもタイトル，アドレス，ユーザ名，パスワードのうちのいずれか一つしか見えない「のぞき窓」として働きます（**図1**）．



〔図1〕アカウント・バンクの表示画面は「のぞき窓」として機能する
アカウント・バンクには，「タイトル，アドレス，ユーザ名，パスワード」のセットが複数登録されている．アカウント・バンクの表示画面に一度に表示できるのは，一つの登録データの一つの項目だけである．

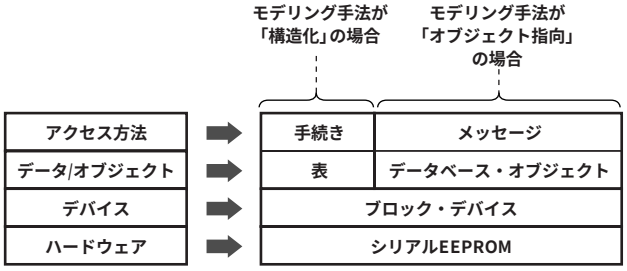
●全体の中の位置づけを再確認する

モデルを作るときは，階層的に抽象化して考えます．階層化することで，各層（ハードウェアやソフトウェア）を部品として利用しやすくなります．階層化モデルの例としては，ISO（International Organization for Standardization）のOSI（open systems interconnection）ネットワーク参照モデルが有名です（**図2**）．インターネットはこれに類似した階層モデルをもとに成り立っており，物理的な違い（無線，有線など）を超えて同じアプリケーション・ソフトウェアを使えるようになっています．

シリアルEEPROMに対するアクセスを階層的にモデリ

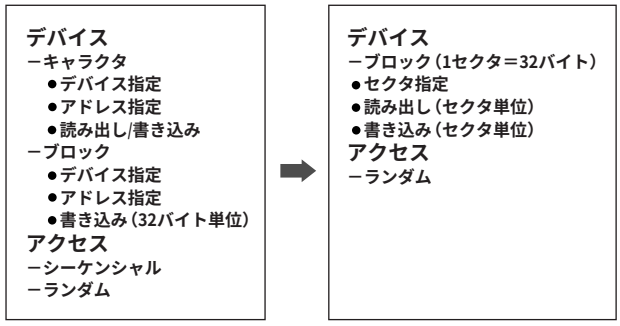
レイヤ7	アプリケーション層
レイヤ6	プレゼンテーション層
レイヤ5	セッション層
レイヤ4	トランスポート層
レイヤ3	ネットワーク層
レイヤ2	データリンク層
レイヤ1	物理層

〔図2〕OSIネットワーク参照モデル
ISOの定義したネットワーク参照モデル．なお，インターネットに使われているプロトコル群は，もう少し階層を大ざっぱに分けたモデルをもとに成り立っている．



〔図3〕シリアルEEPROMに対するアクセスの階層モデル

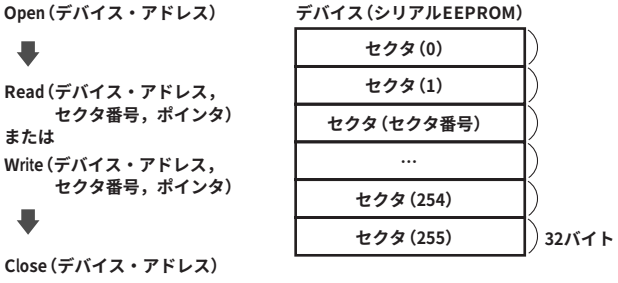
シリアルEEPROMに対するアクセスは，ハードウェア，ハードウェアに依存しないデバイス，データ(表形式)，アクセス方法(手続き)という複数の層に分けて考えることができる．それぞれの層に応じたモデルが存在する．なお，今回はモデリング手法として構造化手法を用いている．



(a) シリアルEEPROMの機能　　**(b) アカウント・バンクにおいてデバイスとして提供する機能**

〔図4〕シリアルEEPROMで利用する機能を整理する

ハードウェアの持っている機能を抽象化し，デバイス機能として上位層に提供するために図にまとめてみる．まず，UNIXのデバイス(入出力機器)モデルを借用して，シリアルEEPROMが持つ機能(入出力の単位，アクセス手段)を(a)にまとめた．また，その中で今回使用する機能を(b)にまとめた．



〔図5〕APIのモデリング

シリアルEEPROMを使い始めるときはOpen関数を使い，シリアルEEPROMを指し示すデバイス・アドレスを指定する．読み出すときは，Read関数でセクタ番号を指定する．該当するセクタから読み出した値(32バイト分)がポインタで指定されたアドレスに読み出される．Write関数，Close関数も同様に機能する．

ングすると，**図3**のようになります．最下層の「ハードウェア」は，シリアルEEPROMなどの物理的なハードウェアを示します．ハードウェア層の上には，特定ハードウェアへの依存性をなくして抽象化した「デバイス」があります．デバイス層の上には，今回のデータ格納のためにモデル化した表現形式を表す「データ/オブジェクト」があります．

データ/オブジェクト層とその上の「アクセス方法」の内容は，モデリング手法が何であるかによって異なります(詳しくはp.98のコラム「構造化手法とオブジェクト指向手法のモデルの違い」を参照)．今回は具体的なふるまいを明らかにしたいので，構造化手法でモデリングしました．そのため，データ/オブジェクト層の内容は表形式にしました．

●ハードウェア層とデバイス層のモデリング

この階層モデルについて，もう少し深く見ていきましょう．まず，ハードウェア層にあたるシリアルEEPROMを汎用デバイスとして抽象化し，抽象化した部分をデバイス層として扱えるようにします．ここでは，デバイス・モデルとして「すべてのデバイスをファイルとして取り扱う」というUNIXのデバイス定義を流用しました．

シリアルEEPROMは**図4(a)**のような機能を持っています．キャラクタ・デバイス(1文字単位でアクセスする)としても，ブロック・デバイス(固定長の複数バイト単位でアクセスする)としてもアクセスできます[注1]．アクセス手段としては，シーケンシャル・アクセス(順番に読み書きする)と，ランダム・アクセス(任意の場所を読み書きする)があります．

アカウント・バンクとして情報を記憶する単位は，一つの登録内容の中のさらに一つの項目ごと(32バイト単位)です．したがって，キャラクタ・デバイスとしての機能は不要です．さらに順番にアクセスする必要もないので，シーケンシャル・アクセスも不要です(**図4(b)**)．

●アクセス方法層(手続き)のモデリング

次は，アクセス方法層にあたる手続き，すなわちAPI(application program interface)をモデリングします．APIもUNIXの定義をまねて，Open(デバイスを開くときに使う)，Close(閉じるときに使う)，Read(読み出しに使

注1：代表的なキャラクタ・デバイスとしては，ターミナル，プリンタ，テープ・ドライブが挙げられる．ブロック・デバイスとしては，フロッピ・ディスク・ドライブ，ハード・ディスク・ドライブなどがある．なお，ブロック・デバイスはキャラクタ・デバイスとしてもアクセスできる．