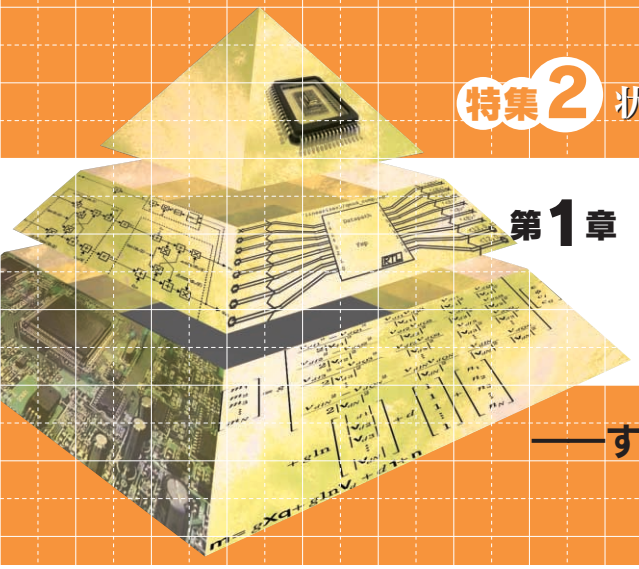


第1章 システムはこうモデリングする！

――すなおな，自然な，むだのない設計を心がけよう

山崎辰雄

機器を設計する際に，ハードウェア開発ではブロック図やタイミング図，状態遷移図，回路図などを利用する．また，ソフトウェア開発ではUML(unified modeling language)としてひとまとめにされている各種表現手法などを利用する．しかし，これらの描きかたを学んでも，モデリングの考えかたや開発手法を学んだことにはならない．本稿では，機器設計におけるモデリングの考えかたやその手順を，具体例を示しながら説明する．（編集部）

1. モデルは何のために作るのか？

思っただけで欲しいものを形にしてくれる「魔法の箱」があればすてきなのですが，残念なことにそのようなものは（まだ）ありません．しかたがないので，ああでもない，こうでもないと考えながら，自分で地道に作っていきます．

犬小屋のように，どのようなものになるのかを頭の中で容易に想像できて，いきなり材料を切ってつなぎ合わせて何とか作れてしまうものならば，それほど苦労はないでしょう．しかし，顧客から依頼されて家を建てるとなると，どのような家を建てるかについて顧客と合意してから作業にとりかからなければなりません．

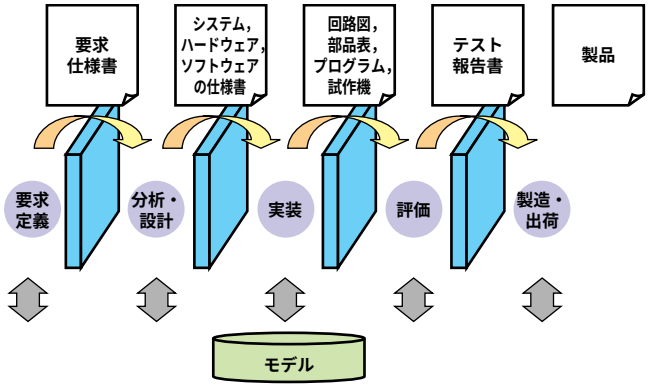
このように，欲しいものを直接形にすることが難しい場合は，中間目標を設けて作っていきます．中間目標を設けることで，ひとかたまりのままでは考えにくい問題が明確になります．

昔，帆船を建造するときはまず縮小模型を作り，船主や設計者，現場監督などの関係者が確認し，合意したうえで実際の船を造る作業に取りかかっていました（図1）．つまり，「船が欲しい→縮小模型を作る→実際の船を造る」という手順を踏んで作業を進めていたわけです．縮小模型を作る過程で，実際に水に浮かべて性能を試します．また，設計変更箇所のチェックはもちろん，いくつの部品が必要か，製作期間はどのくらいかかるか，費用はいくらになるかなど，実際の船を造るために必要なおおよその見積もりも，その縮小模型を参照して得ていたのです．船の建造中も，各部分の担当者の作業が全体の中のどういう位置づけになるか，その作業がほかへどう影響するかなどを確認するために，何度も縮小模型を参照しました．

〔図1〕帆船の建造
昔，帆船を建造するときはまず縮小模型を作り，船主や設計者，現場監督などの関係者が合意したうえで，実際の船を造る作業に取りかかっていた．

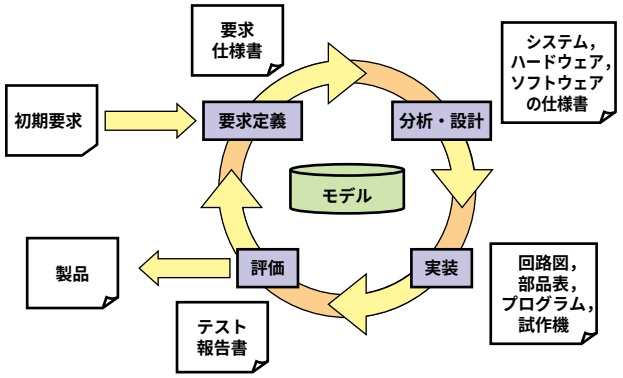
●モデルを使って効率的に作業を進める

今日の製品開発でも，アイデアから製品を一気に作りたいところです．しかし，関係する複数の人々の間で作りたいもののイメージを共有し，複雑な機能を決められた期限までに完成させるとなると，どこから手をつけてよいかわからなくなります．そこで，作業工程を「要求定義」→「分析・設計」→「実装」→「評価」→「製造・出荷」という段階に分けて，それぞれの作業工程の成果物として「要求仕様書」，



〔図2〕ウォータ・フォール・プロセス

各工程できちんとした成果物を作成してから，次の工程へ移る．後戻りが難しい(または許されない)ものに適用する(造船やビルディングの建設など)．モデルがあれば，前工程の完了を待たずに，どのようなものが欲しいのかがわかる．



〔図3〕繰り返しプロセス

完成するまで何度も各工程を回し，製品になるまで繰り返す．要求が絶えず変わり続けるもの，または何度もやり直すことができるものに適用する．モデルはつねに参照・更新される．

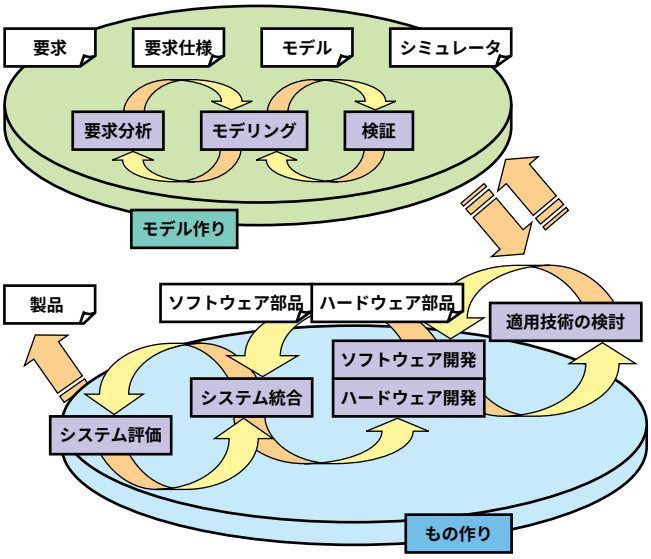
「仕様書(システム，ハードウェア，ソフトウェア)」，「回路図，部品表，プログラム，試作機」，「テスト報告書」，「製品」を作ることにします(図2)．

現在，主流になっている作業の進めかたは，作る製品の特徴によって2通りあります．一つは，橋，船，ビルディングのように一度作り始めたらやり直しのきかないものに使う「ウォータ・フォール・プロセス(図2)」です．もう一つは，製品仕様が開発中に何度も変更されるソフトウェア製品や，携帯電話を代表とする家電製品の開発に使う「繰り返しプロセス(図3)」です．

どちらのプロセスも，モデリング(モデルを作ること)に基づいて異なる作業工程間の意識合わせを行えます．また，みずからの作業内容を前工程の完了を待たずに始めることができます．これを「モデルに基づいた開発手法」と言います(図4)．この手法では，まずモデルを作り，次に具体的なもの作りに取りかかります．モデルは，要求仕様ができた段階から作り始めます．帆船の縮小模型のように，以降の作業段階で参照し，さらに改訂を繰り返しながら使います．

ハードウェアを表現するモデルとしては，ブロック図，タイミング図，状態遷移図，回路図などがあります．ソフトウェアを表現するモデルとしては，近年，以下のような表現手法がUML(unified modeling language)としてひとまとめにされています．

- 静的な構造を表す―― クラス図，オブジェクト図
- ふるまいを表す―― シーケンス図，コラボレーション図，ステート・チャート(状態遷移図)
- 要求定義に使う―― ユースケース図，アクティビティ図



〔図4〕モデルに基づいた開発手法

上の円はモデル作りを，下の円はもの作りを示す．モデルを作成した後に，具体的なもの作り(実装)に入る．モデル完成時点で，どのようなものになるか，どのような技術が必要かを判断する．作り始めてからの見通しの悪さを軽減することができる．

- アーキテクチャを表す―― コンポーネント図，配置図

ただし，これらはあくまでも「お絵描き」の作法であって，設計の考えかたや開発手法を示してくれるものではありません．絵筆が持てるからといってだれでも印象的な絵が描けるわけではないのと同じことです．風景を写生するには，遠近法に相当する考えかたや手法を学ばなければならないのです．ここでは，システム設計に不可欠なハードウェア・モデルやソフトウェア・モデルの作りかた，すなわち「モデリング手法」を解説していきます．