

状態構造に着目した動作仕様化の一技法

3R-1

小山田正史, 岩戸伝一  
情報処理振興事業協会・技術センター

1 はじめに

ソフトウェアの動的なふるまいを対象としたさまざまな設計(仕様化)技法が開発されている。多くは、対象システムを、静的な視点(機能階層と構成要素間の関係)と動的な視点(状態遷移など)として捉えている(例えば, [1])。しかし、システムや構成要素の状態を構造的に捉えているものは少ない。動的な視点から状態構造を捉えてたものとして、文献[2]がある。

本稿では、静的な視点からシステムの状態を構造的に捉えた動作仕様化の技法、およびその応用について述べる。

2 背景

システムの動的なふるまいの説明は、例えば「電話機の(1)受話器を上げて、(2)ダイヤルを回して、…」のように、操作(あるいは制御)の列によって行われることが多い。状態遷移図(表)による記述も、対象の操作(制御)手順を表している。

このような操作列による表現は、(a) 対象の構造を前提として、(b) 構造に対する操作前後の変化分(差分)を表している。操作の列の記述だけを見ても、システムの動作を理解することは難しい。状態(構造)の制御が中心となるシステムでは、状態を操作する手順だけでなく、操作によって変化したシステムの状態(構造)も含めて理解する必要がある。このため、動作設計ではシステムの抽象的な状態構造を含めた仕様化技法が望まれる。

3 状態指向による動作仕様

表1: 状態構造

コンストラクタ	意味
プリミティブ	インスタンスの状態値
構成的メンバ	インスタンスの状態は、メンバからなる状態ベクトル。
択一的メンバ	インスタンスの状態は、メンバのどれか一つの状態。

状態構造の枠組: 動作対象の状態の枠組は、表1に示すコンストラクタを使った構造アイコンタイプとして定義される。また、構造アイコンタイプは名前構造タイプ名を持つことができる。さらに、構造ア

アイコンタイプはその要素にシンボルやテキストなどの視覚的な表現を割り当てることにより、動作対象の構造を論理的かつ視覚的に表現できる[3]。図1に電話機の定義例を示す。

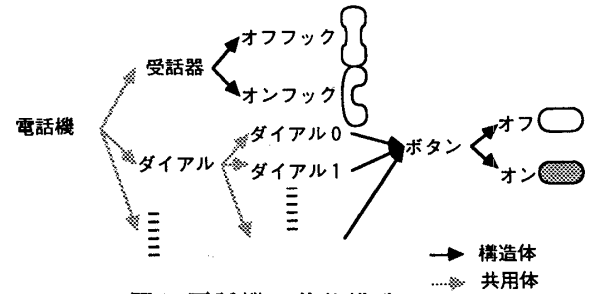


図1: 電話機の状態構造例

仮想的な動作対象集合: 一般に、動作対象の動作は、その外部環境の相互作用によってモデル化される。外部環境として一つにまとめるのではなく、外部環境の具体的にあるいは仮想的に特定できるいくつかの動作対象(仮想的な動作対象)の集まりとして捉える。

例えば、電話機の仮想的な動作対象は、「ユーザ」と「交換機」である(図2)。

状態構造のスコープ: 電話機では、受話器やダイヤルの状態構造を制御するのは「ユーザ」であり、呼び出しレベルを制御するのは「交換機」である(図2)。「ユーザ」、「交換機」は、電話機のすべての状態構造を制御できるわけではない。動作対象の状態構造に対して、仮想的な動作対象が制御できる範囲をスコープと呼び、スコープと仮想的な動作対象の関係を制御関係と呼ぶ。

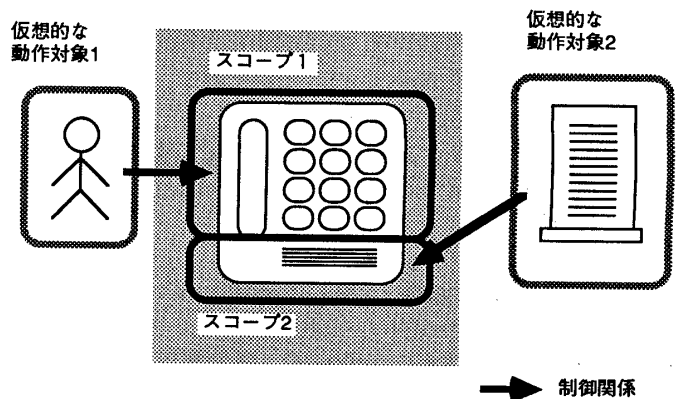


図2: 状態構造のスコープ

**動作対象間の関係:** 動作対象間の関係は、動作対象のもつ仮想的な動作対象とシステムを構成するほかの動作対象との結合関係によって示される。動作対象「電話機」では、仮想的な動作対象「ユーザ」はシステム外部環境に、「交換機」は具対的な交換システムに結び付けられる(図3)。

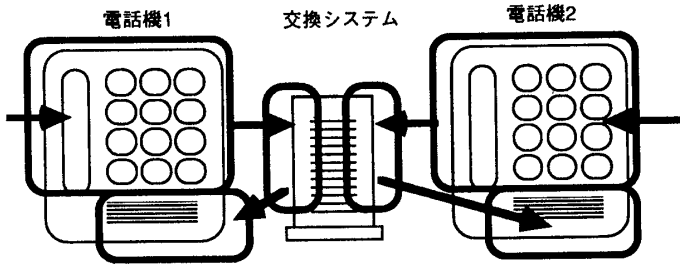


図3: 動作対象間の制御関係

**動作シナリオ:** システムのさまざまな動作(例)について、システムを構成する動作対象の状態構造を使い時系列として表す(図4)。このように記述されたシステムの一連の動作を動作シナリオと呼び、各時点でのシステム全体の状態をフレームと呼ぶ[3]。

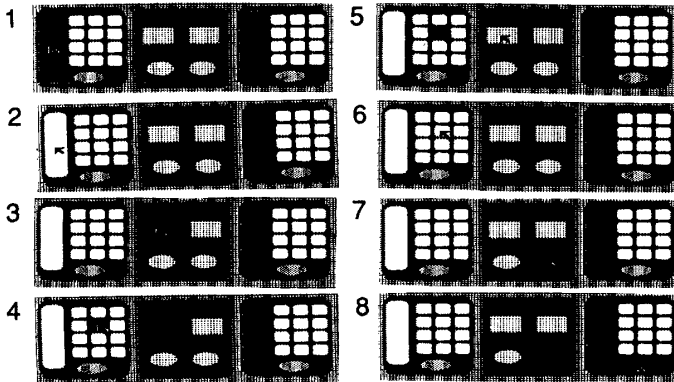


図4: 動作シナリオの例

本技法では、従来の技法で採り入れられていた動作対象間の関係(メッセージの転送方向)だけでなく、動作対象の状態構造とその制御関係を静的な視点から表現することができる。

4 本技法の応用

動作シナリオに対する基本的な支援機能は文献[4]で述べた。本節では、特に状態構造の記述に特徴的な支援機能について述べる。

4.1 動作シナリオの作成支援

- (1) **状態構造の修正:** 択一的なメンバを順次表示することができる。
- (2) **編集対象のナビゲーション:** 制御関係を使って、次の時点で状態変化可能な動作対象とその状態構造を示すことができる。
- (3) **状態変化のチェック:** 制御関係に矛盾する状態変化を検出することができる。

4.2 動作シナリオからメッセージシーケンスへの変換

動作シナリオでは、システムを構成する個々の動作対象の状態を時系列的に表している。一つのシステム動作として理解する場合、状態変化に対する動作対象間の係わり(因果関係)を知る必要がある。以下に、動作シナリオから状態変化の因果関係を得る手順を示す。

- (1) **状態変化の導出**  
動作シナリオの前後するフレームの動作対象の状態(構造)を比較し、等しくなければその動作対象に状態変化があったとする。
- (2) **状態変化の因果関係の導出**  
動作シナリオの各状態変化について、状態変化のあった状態構造を制御する動作対象を制御関係により求める。制御した動作対象のもっとも最新の状態変化を、その状態変化の原因とする。制御した動作対象がシステム環境の場合は、システムの外部に原因があったものとし、動作シナリオ内部の因果関係には含めない。

この因果関係が、異なる動作対象間で成り立つ時、メッセージ転送が行われたことになる。一方、同じ動作対象間で成り立つ時は、動作対象の内部オペレーションが実行されたことになる。

- (3) **転送メッセージ名の生成**  
状態変化した部分の状態構造に付けられた名前のパスに着目する。メッセージ転送がある時、このパス名は受信したメッセージの名前とし、内部オペレーションの場合は、オペレーション名とする。例えば、受話器が上がった時は、“電話機受話器オフフック”というパス名がメッセージ名となる。

本手順を図4に適用した例を図5に示す。

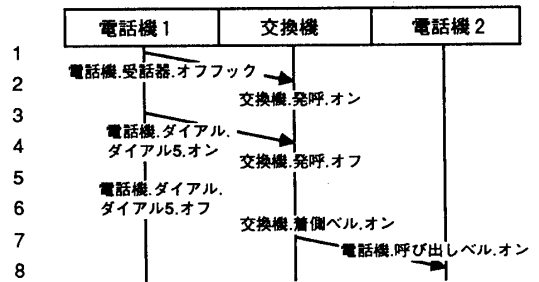


図5: メッセージシーケンス変換例

5 おわりに

以上、状態の構造を考慮した動作仕様化の技法について述べた。本技法は、SNAPSHOT 設計技法[3]の一部として導入されている。

参考文献

- [1] CCITT Recommendations Z. 101 - Z. 104, 1984.
- [2] D. Harel, et. al. "STATEMATE: A Working Environment for the Development of Complex Relactive Systems," 10th ICSE, 1988.
- [3] 小山田, 岩戸「リアルタイムソフトウェア向け設計支援の提案- 動作設計技法の概要-」, 情処全大1989.
- [4] 岩戸, 小山田「リアルタイムソフトウェア向け設計支援の提案- 支援システムの概要-」, 情処全大1989.