

有向グラフの階層に基づく  
対話型応用ソフトウェアの仕様記述の一方法

2S-3

西尾 高典<sup>†</sup>      千吉良 英毅<sup>††</sup>      今城 哲二<sup>†</sup>

<sup>†</sup>(株)日立製作所 システム開発研究所    <sup>††</sup>同 情報システム工場    <sup>†</sup>同 ソフトウェア工場

1. はじめに

OAシステムに搭載される応用ソフトウェアに対して高度のユーザ操作性が要求されるにつれ、対話型ソフトウェアの生産技術の向上が重要な課題となってきた。一つのアプローチとして、対話型ソフトウェアの仕様記述からプログラムを自動生成するという方法がある。一般にこの種の方法では、仕様の了解性と汎用性を両立させることが問題である。両者を満足する仕様記述法を確立するためには、その基礎となるモデルを、対話処理に十分適合したものにすることが必要である。本稿では、このようなモデル及び仕様記述法を提案する。

2. 基本的な考え方

図1に示す具体的な伝票処理を例にして対話処理を考える。システムメニューから開始して、売上メニュー、売上傳票へと遷移しながら、個々の画面処理を実行する。画面処理は、画面表示に始まり画面消去で終了する。一般に画面は幾つかのブロックから構成され、画面処理中に各ブロックの処理を実行する。ブロックは、オペレータが同時にアクセス可能な(活性化された)領域の単位である。例えば、売上傳票は、見出し、明細行、集計というブロックから構成される。見出しから開始して、明細行、集計の順に次々とブロックを遷移しながら、個々のブロック処理を実行する。ブロック処理では、構成する項目データの一括送受信が実行される。個々の遷移は、図中に示すとおり、PFキー等の制御キーの押下によって指示される。

売上傳票および得意先台帳保守間の操作PF9、PF8や明細行の操作ENT(送信)、BT(後退)は、同一形式の画面/ブロック間の遷移である。これは、同一形式の画面やブロックの定義(クラス)から次々と実体(インスタンス)が生成され、インスタンス間で遷移が行われていると解釈される。あるインスタンスから別のインスタンスに遷移する場合、それぞれの基となるクラス間に時間的な半順序関係があると解釈される。又、対話、画面、ブロックの個々のクラスは、この順に親子関係にある。一つの対話クラスは幾つかの画面クラスを、また個々の画面クラ

スは幾つかのブロッククラスを子としてもつ。子は親にたいして半コルーチンである[1]。あるクラスのインスタンスは初期処理実行後、子のインスタンス生成、呼出(call)を次々と実行し、最後に終了処理実行後、親に戻る(end)。祖先の遷移が指定された場合は、処理を停止(suspend)して親に制御を戻す(detach)。対話モデルは、このような対話処理の形式的記述の基礎となるモデルである。

3. 対話モデル

(1)モデルの構造

対話モデルとして、「閉路をもたない有向グラフ(Directed Acyclic Graph:以下dag)[2]の階層」を導入する。dag  $G = \langle V, E \rangle$ 上の個々の頂点 $v \in V$ は、他のdag  $G'$ と結合することができる(図2)。頂点 $v \in V$ は、画面やブロックなどのクラスを、弧 $e \in E$ はクラス間の時間的な半順序関係を意味する。頂点と他のdagとの結合は、画面とブロックのようなクラス間の親子関係を意味する。他のdagと結合しない最下位の頂点は、送受信を実行するブロッククラスを示す。

(2)モデル上の操作

対話モデルの操作は、 $G = \langle V, E \rangle$ 上での頂点 $v_i \in$

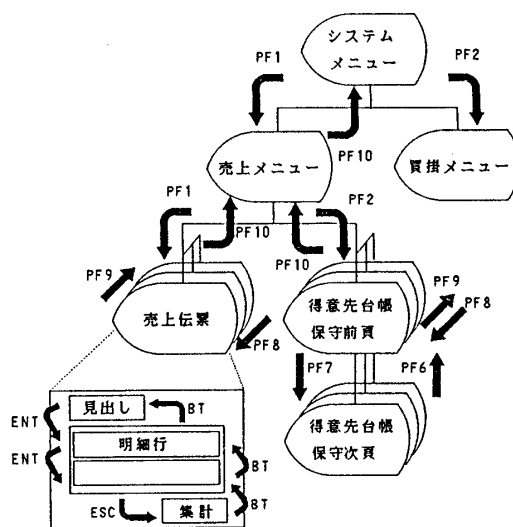


図1. 対話処理の例

Specification Technique for Interactive Application  
Software based on Multi-layered Directed Acyclic Graph

Takanori NISHIO<sup>†</sup>, Eiki CHIGIRA<sup>††</sup>, Tetsuji IMAJO<sup>†</sup>

<sup>†</sup>Systems Development Laboratory, Hitachi Ltd., <sup>††</sup>Systems Design Works, <sup>†</sup>Software Works

$V$  から頂点  $v_2 \in V$  への訪問である。従って操作は、操作対象 (dag) と訪問の仕方の対で指示される。訪問の仕方には、弧  $e \in E$  の方向に対して正向き (次) と逆向き (元) の訪問、その頂点のインスタンス間相互の訪問 (後, 前),  $G$  の上限と (もしあれば) 下限への直接訪問 (先頭, 末尾),  $G$  と結合した上位頂点への戻り (終了) が考えられる。

対話モデルに対して操作を施した結果は、ある最下位頂点から別の最下位頂点への訪問である。これは、最下位頂点から操作指示が指定する  $G = \langle V, E \rangle$  に到る dag 階層の上方へのたどり、操作指示の指定する訪問の仕方に従った頂点  $v_1 \in V$  から頂点  $v_2 \in V$  への訪問、頂点  $v_2$  から最下位頂点に到る dag 階層の下方へのたどりから構成される。dag 階層の下方へのたどりは、個々の頂点の示すクラスのインスタンスの一連の呼出を意味する。dag 階層の上方へのたどりは、これらのインスタンスの一連の戻りを意味する。各 dag は常に唯一の印付き頂点をもつ。これは、そのクラスのインスタンス処理の停止点を意味する。初期的には dag の上限頂点が刻印されており、訪問元から訪問先へと次々と印を付け替える。

#### 4. 対話モデルの特徴

対話処理記述に一般に広く用いられている状態遷移モデルと比較しながら、対話モデルの特徴を述べる。

対話モデルの最も大きな特徴は、状態遷移モデルでは弧 (「遷移」) が処理、頂点が「待ち」状態を表すのに対して、グラフの頂点がクラスを表しているという点である。

一般に対話処理は「受信」「入力データ処理」「遷移先決定」「出力データ作成」「送信」のサイクルである。状態遷移モデルにおける一つの遷移 (入出力変換) は、「受信」から「送信」までの処理に対応する。「遷移先決定」の直後に画面が切り替わるので、一つの遷移が、遷移元、遷移先の画面処理を合わせ持つことになる。これは、仕様記述を著しく複雑にする。対話モデルでは、「送信」/「受信」の対が、下位 dag 訪問/戻りの対と適合するので、頂点を「出力データ作成」から「遷移先決定」という画面処理と対応づけることが可能となり、上記の問題は解決される。

状態遷移モデルでは、遷移が受信/送信の単位なので、ブロック以外の遷移をモデル化することは出来ない。これは、画面処理、ブロック処理等が一つの遷移に混在することを意味し、仕様の簡潔性を損なう。対話モデルの場合、一つの頂点の詳細化は、処理のトップダウン分割と対応する。従って、画面処理、ブロック処理等をその階層毎に完全に分離して記述することができる。

#### 5. 仕様記述とプログラム生成

仕様は、対話モデルの概念を構文的に記述するための言語である。対話モデルにおける頂点は、画面やブロッ

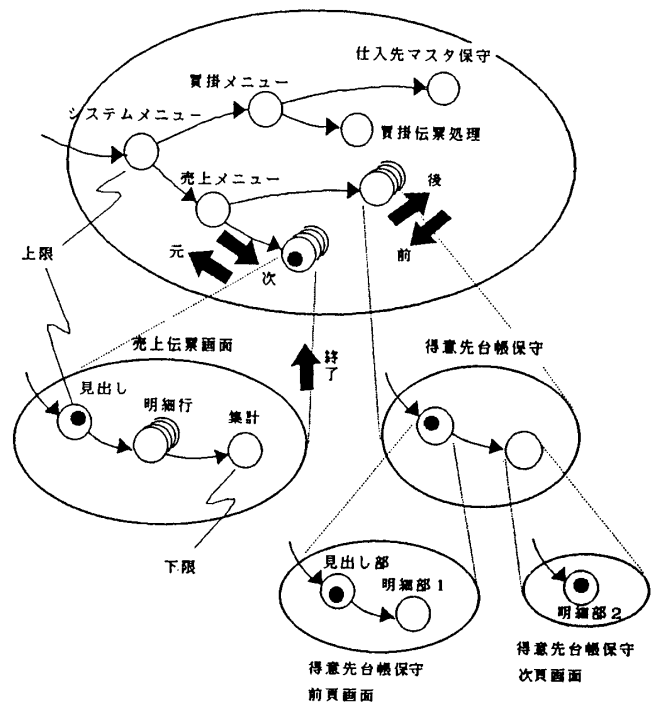


図2. 対話モデルの構造と操作

クなどのクラスとして定義される。対話モデル上の頂点間の構造的な関係は、クラス間の半順序関係、親子関係によって定義される。その他、そのクラスに許される操作と制御キーとの結合、初期処理、終了処理が、クラスの属性として定義される。各クラスに共通に与えられる処理ロジックの枠組に、仕様で定義した初期処理、終了処理、そのクラスの物理的な意味付け (画面、ブロック等) に応じた固有処理を埋め込むことにより、対話型ソフトウェアが生成可能である。

#### 6. おわりに

○ A システムに搭載される対話型応用ソフトウェアの仕様記述法及びプログラムの自動生成法を提案した。dag の階層を対話モデルとして導入し、この上で対話処理におけるオペレータの操作を定式化したことに特徴がある。複雑な遷移処理を簡潔に仕様可能であると同時に、従来別々に扱われてきた画面間の遷移処理、画面内の項目遷移処理などを、統一的に扱うことが可能となる。

#### 参考文献

- 1] Dijkstra, E.W., Hoare, C.A.R., Dahl, O. [1972]. Structured Programming, Academic press. 邦訳: 野下, 川合, 武市. 構造化プログラミング, サイエンス社, 1975.
- 2] Aho, A.V., Hopcroft, J.E., Ullman, J.D. [1983]. Data Structure and Algorithms, Addison-Wesley. 邦訳: 大野. データ構造とアルゴリズム, 培風館, 1987.