

## プロセス統合に用いられるユーザインタフェースの設計

満田 成紀 鯉坂 恒夫 松本 吉弘

京都大学工学部

統合型 CASE 環境におけるプロセス統合サービスの実現を目的とし、環境と人間の界面となるユーザインタフェース (UI) の設計方針を与える。この方針は、UI の中で使用される情報の分類を基本として考えられている。UI 設計者は、ウィンドウを構成する表示領域とその間にある制約を、物理モデルとして設計する。また、ツールへの入出力の流れ (= プロセス) を論理モデルとして設計する。論理モデルは、イベントと状態、それらの間にある因果関係から成る。UI ライブラリは、二種類のモデル間の情報変換を実現するものである。

## Design of User Interfaces for Process Integration

MITSUDA Naruki AJISAKA Tsuneo MATSUMOTO Yoshihiro

Faculty of Engineering, Kyoto University

Yoshida honmachi, Sakyo-ku, Kyoto 606-01, JAPAN

This paper describes a design strategy for user interfaces which provide process integration services in a CASE environment. This strategy is based on a classification of information used by user interfaces. At the physical level, the UI designer models structures of presentation areas in a bitmap window and constraints among them. At the logical level, he makes another model which shows a process of input and output of his tool. This model includes events, states, and the relation of cause and effect on them. The UI library should provide a mapping mechanism between tow levels.

## 1 はじめに

統合型 CASE 環境は、ソフトウェア生産の支援を目的とし、多数のツールやサービスが有機的に結合されたものである。この時、ツールやサービスに対して、統合を実現するようデザインされたインタフェースが要求される。

要求されるインタフェースを規定するために、統合型 CASE 環境の参照モデルとして、図 1 に示すトースターモデルが ECMA/NIST によって提唱されている [1]。ここでは CASE ツールを統合するためのサービスとして、データ、プロセス、制御の三つの統合に着目し、それらとツールとの境界面におけるインタフェースを規定することを要求している。

データおよび制御統合については、これまでに多くの研究がなされ、サービスやインタフェース規格がいくつか提案されている。本研究は、比較的研究が遅れていると思われるプロセス統合について、その背景にあるユーザインタフェースも含め、サービスの構成要素やツールインタフェースの分析、設計を行うものである。

## 2 プロセス統合サービスの構成

ここでは、CASE 環境において利用される情報の分類をもとに、統合サービスの構成を考える。プロセス統合サービスについては、その背景にあるユーザインタフェースの機能をもとに、物理モデルとプロセスモデルに分解する。プロセス統合とは、プロセスモデルの構成要素間に存在する制約を管理することに相当する。

ソフトウェアプロセスの場合、人間の思考活動(プロダクトを対象として一人で行う思考活動や、他人とのコミュニケーション)が大きなウェイトを占めるが、ここでは人間の活動そのものにはふれず、それが CASE 環境で支援される界面に注目している。つまり、思考活動の結果が環境に反映される際の現象を切り出し、環境の側から見た(人間の活動をカプセル化した)プロセスのモデルを与えようとしている。

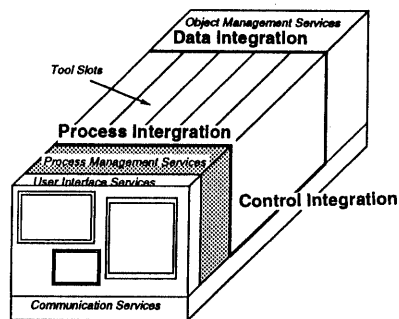


図 1: 統合型 CASE 環境の参照モデル

### 2.1 情報の分類とサービスの構成

トースターモデルによって明らかとなった統合サービスを設計する上で重要なのは、それが特定のツールに依存するものではなく、汎用でなければならないということである。汎用の鍵となるのは、いかにしてサービスの入出力として要求されるべき情報を見究め、それを抽象化するか、ということである。つまり、サービス内で利用される情報を、それが持つ意味をもとに分類することである。

例えば筆者らは、CASE ツール間で相互交換される情報の分類をもとに、ネットワークエディタを作成するための汎用モジュールを実現した [2]。

プロセス統合サービスの場合、ユーザインタフェースの提供すべき機能をもとに、そこで利用される情報を分類することができる。ユーザに近い部分では、ウィンドウシステムによって提供される物理的構成が定められており、そこで用いられる情報とは物理的属性のことである。

一方、ツールに近い部分では、ツールに対する入出力の持つ論理(=プロセス)が情報として定義される。つまり、ユーザインタフェースにおいて利用される情報は、物理的情報と論理的情報の二つに分類することができ、このとき、UI の役割とは、物理 / 論理間で情報変換を行うことであると言える。

以上の分析により、プロセス統合サービスを

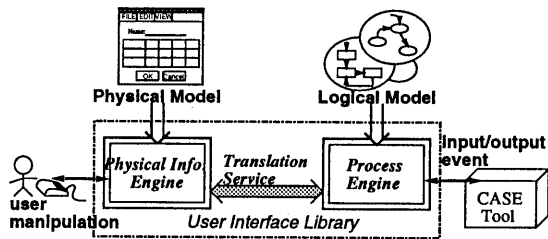


図 2: プロセス統合サービスの構成

実現するには、図 2 に示すように、物理エンジンとプロセスエンジンを用意することが有効であると言える。

## 2.2 UI の物理モデル

現在では、UI は X-Window のようなウィンドウシステム上で実現されることがあたりまえとなっている。したがって、UI の物理的側面はウィンドウシステムによって規定される。

物理的側面だけに注目すれば、UI の設計とは、表示領域の物理的配置を決定することである。したがって UI の物理モデルを作成するためには、表示領域の構成とその間の制約を定義することが必要となる。これは表示フォーマットと呼ばれるものを設計することに相当する。

## 2.3 UI の論理的モデル

UI によって実現される論理とは、ツールに対する入出力の流れであり、これはプロセスであると言える。論理であるから、一般のソフトウェア同様にデータ構造、データフロー、状態遷移の観点からモデル化することが可能である。

前にも述べたように、UI の役割とは物理 / 論理間の情報変換であるから、物理的モデルの構成要素と論理的モデルの構成要素の間で対応付けがなされる必要がある。入出力パラメータである論理データには、一つ (または複数) の表示領域が割り当てられ、データの値をもとにして表示領域の物理的イメージが生成される。ま

た、状態遷移を引き起こすイベントのいくつかは、マウスボタンなどの物理的属性の変化に対応付けられる。

## 2.4 プロセス統合の実現

プロセスを統合するとは、個々のツールの立場でデザインされたプロセスモデルに対して、CASE 環境としての一貫性を保持するための制約を与えることである。具体的には、プロセスモデルの各構成要素について他の構成要素との依存関係を記述し、それを管理することである。例えば、イベントの生起について順序関係を与えることは、ツール操作のプランニングを行うことに相当する。

本研究においてプロセス統合を実現する UI とは、物理 / 論理を切り分けることで人間による活動をカプセル化し、論理モデルを構成する要素に対して制約を与える手段を提供するものである。人間の活動も含めた一様なモデル化を提供することによって、複数のユーザやツールによる協調的活動 (= パフォーマティブ) も、いくつかのプロセスモデルにまたがる制約として容易にモデル化できる。

## 3 考察

これまで、プロセス統合を目的とし、UI の中で操作される情報の分類をもとに、UI の構成を考えてきた。物理モデルとプロセスモデルに分け、各々を支援するサービスを統合する形での、UI ライブラリの設計が重要である。その実現の鍵となる技術について考察する。

### 3.1 物理モデルの制約管理

標準的なウィンドウシステムを考えた場合、物理モデルを構成する要素にはウィンドウ内の表示領域がある。個々の表示領域は、それぞれ属性として、位置、形状、サイズ、ビットイメージを持つ。また、マウスやキーボードなどの入力装置も構成要素であり、各々の属性を持つ。

物理モデルにおける制約とは、位置やサイズといった属性値を制御するものである。例えば、表示領域の位置とマウスの状態を関係付けることで、領域のドラッグ & ドロップが可能となるし、またこれを禁止することもできる。

制御される対象である属性の種類は固定であり、また、これらの属性が持つ物理的意味も明確である。したがって、これらの管理サービスを設計するのは、比較的容易であろう。

### 3.2 プロセスモデルの制約管理

プロセスモデルは、ツールに対する入出力の持つ論理を、ツール設計者が自由にデザインしたものである。入出力パラメータをデータ、その値の変遷をアルゴリズムと考えることで、一般のソフトウェア同様のモデル化が可能である。ただし、データの構成(静的側面)よりも、変遷の過程(動的側面)に重点を置いたモデルである。

論理の動的側面を表現するためによく用いられるのが状態遷移モデルであり、プロセスモデルの場合もこれを利用できる。状態とイベントという抽象的な概念を持ち込むことで、プロセスの中に潜む因果関係を明確に示すことができる。プロセス統合とは、この因果関係に焦点を絞った統合である。

イベントと状態の概念を持ち込むことで、プロセスモデルにおける制約と言うものも明確となる。考えられるのは、イベント-イベント間、イベント-状態間、状態-状態間の制約である。例えば、イベント-イベント間の制約には、構成関係(複数の副イベントへの分割)、順序関係、排他関係などが考えられる。

### 3.3 物理 / 論理情報変換

UI ライブラリとしては、物理モデルとプロセスモデルを各々管理するだけでなく、情報変換を行うことで二つのモデルを連結する必要がある。すなわち、それぞれの構成要素間の対応関係を明確にし、一貫性を管理する必要がある。

物理モデルの表示領域に対応するのが、プロ

セスモデルの入出力パラメータである。その表示領域には、パラメータの値をもとにして作られるビットイメージが表示される。この時、どのようなビットイメージが作られるかは、表示領域の種類によって異なる。整数値のパラメータに対して、数字が表示されるかもしれないし、スライディングバーが表示されるかもしれない。

プロセスモデルのイベントの多くは、物理モデルにおける入力装置の状態に対応する。例えば、マウスの状態をもとに「表示領域 A における左ボタンのプッシュ」を意味するイベントの生成を考えることができる。ただし全てのイベントがそうとは限らない。ツールからの出力要求もイベントであるし、因果関係を表すために便宜上導入された仮想のイベントもあり得る。

結果として、図2に示した物理エンジン-プロセスエンジン間で必要な変換サービスとは、

- パラメータのビットイメージ化
- 入力装置状態のイベント化

の二つであると言える。

## 4 おわりに

本研究は、CASE 環境で扱われる情報を、それが持つ意味をもとに分類し、プロセス統合サービスの設計に応用したものである。これによって、サービスを提供する UI ライブラリの汎用性、モジュール性の向上が期待できる。今後は、より具体的な設計を進め、実用性の高いライブラリを実現することを考えている。

## 参考文献

- [1] ECMA and NIST. "Reference Model for Frameworks of Software Engineering Environments". Technical Report ECMA TR/55 2nd Edition, NIST Special Publication 500-201, December 1991.
- [2] Naruki Mitsuda, Tsuneo Ajisaka, and Yoshihiro Matsumoto. "A Semantic-Directed Graph Editor on PCTE". In *JCSE '93*, 1993.