

適応型ユーザインタフェース（AUI）の構築

2J-3

—オブジェクト指向タスクアナリシス—

瀬野尾 健 長崎 等 東 基衛

早稲田大学理工学部

1 はじめに

現在、対話型システムにおけるユーザインタフェース（UI）の重要性が大きく認識されている。UIは、ユーザのタスクを効率よく達成させるための技術であり、設計者はユーザを支援するタスクについての十分な理解が必要である。

本研究では、ユーザのタスクを分析し、タスクの中の操作と対象を抽出し、関係付けるための分析モデルを提供することを目的とする。その中で、オブジェクト指向の概念を導入した分析方法論を示し、効果や利点についての考察を行う。

2 問題点

従来のタスク分析では、高レベルのタスクをサブタスクへと分解し、タスク階層を推論していく過程でタスクの対象と操作の関係付けが行われていた。このようなタスク分解の問題点として、以下のようなことがあげられる。

- タスクの対象と操作の関係付けの具体的な指針がなく、状況に依存している。
- タスクの対象と操作の関係が、タスク階層によって制約を受けてしまう。
- 抽出したタスクの十分性の判断が難しい。

これらの問題点の根本的な原因として、実世界のタスクを想定するだけでは、表現しきれていないことにある。

3 分析プロセスとタスクモデル

3.1 分析プロセスの概要

提案する分析プロセスとタスクモデルの関係を図1に示す。提案するタスクモデルは、ユーザのタスクを実世界の事象に忠実に分析するためのモデルである。実世界のタスクからオブジェクトを抽出し、その構造や性質、関係に着目することによって、必要な操作を割り当てる。分析プロセスは次の4つの部分から構成されている。

- (1) オブジェクトの抽出
- (2) 属性の定義
- (3) オブジェクトの組織化
- (4) アクションの定義

3.2 タスクの構成要素

以下では、分析対象となるタスクの構成要素を示す。

- オブジェクト
オブジェクトは、タスク領域に従属した事象の集合を抽象化したものとする。
- 属性
属性は、オブジェクトの構造や性質を記述するものである。
- アクション
アクションは、タスクの対象への操作の選択とその操作の遂行を意味する。すなわち、タスクの操作部分である。
- 関係
オブジェクトの組織化や関係付けの機能は、概念的包含関係（is-a関係）や構造的集約関係（has-a関係）を含むやり方で扱われる。

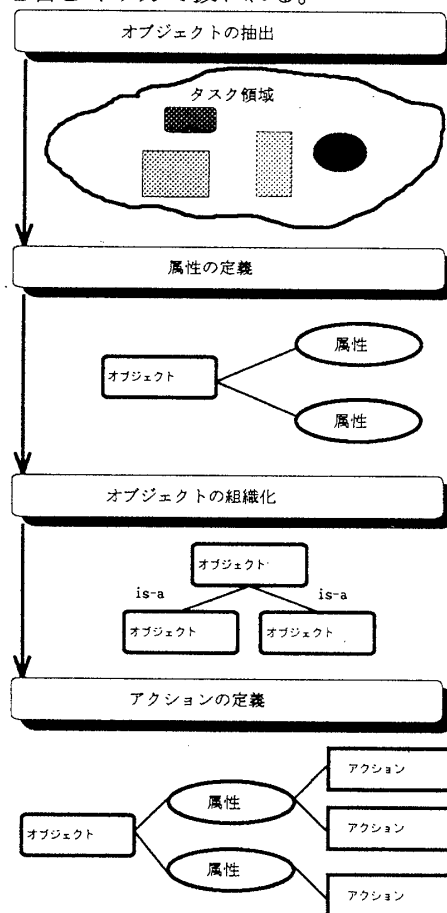


図1 分析プロセスとタスクモデル

4 分析方法論

提案するタスクモデルの記述方法を図2に示す。

```

object : オブジェクト名

attribute:
  属性名 属性の型
  属性名 属性の型
  . . .
action:
  制約; アクション名 (引数)
  制約; アクション名 (引数)
  . . .

```

図2 提案モデルの記述

4.1 オブジェクトの抽出

まずはじめに、ユーザのタスク領域を対象として、さまざまな事象を抽象化し、オブジェクトとして抽出しなければならない。それらのオブジェクトは、ユーザがアプリケーションを用いて生成、編集、削除する対象である。そのオブジェクトに対して、一意的な名前をつける。何をオブジェクトとするかは、対象とするタスク領域に依存している。

4.2 属性の定義

抽象化したオブジェクトの属性とその型を定義する。オブジェクトが保有する構造や性質を決定する情報を記述する。また、あるオブジェクトの属性に他のオブジェクトを埋めこむことによって、構造的集約関係 (has-a関係) を表現することができる。属性は抽出したオブジェクトを組織化するときや、オブジェクトに対する操作を定義するときに必要な概念である。属性は、オブジェクトとアクションを関係付けるための、パラメータとしての役割を果たす。

4.3 オブジェクトの組織化

属性が定義されたオブジェクトは、継承を用いて概念的包含関係 (is-a関係) が定義される。オブジェクトに対してはさまざまな属性が定義されているが、それらは異なるオブジェクト間で共通したものが定義されている可能性がある。そうした場合、上位のオブジェクト (親オブジェクト) に属性やアクションを定義しておき、下位にあるオブジェクト (子オブジェクト) は上位のオブジェクトのものを継承すればよいので、新たに定義する必要がない。また、下位オブジェクトに固有の属性やアクションは、新たに追加することができる。

4.4 アクションの定義

アクションはオブジェクトの属性に依存するので、最後にアクションを定義する。アクションは、オブジェクトの属性を操作するものなので、アクションには操作する属性を引数としてともなう。アクションは暗黙の引数として、そのオブジェクト自身を含

んでいる。さらに、アクションを実行するための事前条件があれば、制約として合わせて記述する。

5 適用事例

以下にワープロに対して適用し、抽出した「パラグラフ」オブジェクトの記述の一例を図3に示す。

```

object : Paragraph

attribute:
  point    Location
  width    Integer
  length   Integer
  content  Content
action:
  ; select (point)
  ; indent (width)
  ; delete (content)

```

図3 「パラグラフ」オブジェクトの記述例

6 利点

本研究の提案モデルによるタスク分析には、以下のような利点がある。

- UIの設計にありがちな、ユーザの視点と設計者の視点のギャップを、オブジェクトを通じて軽減することができる。
- オブジェクトの組織化によって、タスクに関する知識を体系的に分類することができる。
- 一度定義しておけば再利用することができる。
- ユーザタスクの全体像を把握することができる。
- 分析結果の変更や拡張が容易である。

7 考察および今後の課題

従来タスク分解による方法と比較して、改善された点を考察する。

タスク分解では、タスク階層の一部を変更すると、その影響がその下位の階層にまで及んでしまった。しかし、提案モデルによる分析では、アクションがオブジェクト単位にカプセル化されているので、あるオブジェクトにおいて定義を変更しても、他のオブジェクトには影響が及ばないで済むようになった。

また、非常に困難であったことは、分析の出発点であるタスク領域従属のオブジェクトの抽出である。何をオブジェクトと見なすかは、分析者の抽象化能力に大きく依存してしまう。これに対処するためには、分析事例を増やしていくことにより、その結果を再利用していくことが賢明である。

今回はタスクの静的な側面のみを分析するためのモデルを提案した。今後、タスクの動的な側面、すなわち、アクションの実行順序を考えることによって、システムの対話部分の設計に応用していくことを考えている。