

6K-1

アイコンシステムの開発

山口真悟 棚木浩二 西村世志人 田中稔

山口大学 工学部

1 はじめに

近年のユーザインターフェイスの研究により視覚的情報の有用性が報告され、それを利用したシステムが開発され成果を挙げている[1],[2]。しかしこのようなシステムの開発はさらに多くの時間と費用を必要とする結果を招いた。このような理由からプログラミングを支援するツールがシステムデザイナーにより要求されている。

本研究ではアイコンをオブジェクト指向に基づいたカプセル化されたデータと機能として定義し、ユーザの作業空間をコンピュータ上にアイコンを用いて模擬したものをアイコンシステムと定義する。アイコンの集合及び操作は現実世界での”もの”とそれに対する”操作”を模擬したものでありユーザに直観的な情報を与えるため、高いユーザインターフェイスを得ることができる。このようなアイコンシステムを構築するための環境とそのインターフェイス機能及び内部メカニズムについて報告する。

2 アイコンシステム

2.1 アイコン

アイコンシステムにおける操作対象の基本単位はアイコンである。アイコンは内部記述と外部記述の2つの部分から構成される。外部記述は”もの”に対応した視覚的情報である絵シンボルをもつ。一方内部記述はオブジェクトのデータとメソッドをもつ。アイコンのもつデータは情報隠蔽されておりデータは自身をもつメソッドによってのみ操作することができる。アイコンが交信可能なメッセージはリストに登録され、アイコンの実行に用いられる。メッセージリストは active message と passive message の2つのリストがあり、各アイコンは passive message に対応づけられたメソッドのみを有する。このようにつくられたメッセージリストを用いてアイコンのデータ操作を行う。アイコンシステム上のアイコンのふるまいは、他のアイコンと重ね合わせられた時に、互いにメッセージを

交換することによって決定される[3]。

2.2 アイコンシステム

アイコンシステムはアイコンの集合を扱う作業空間であり、このようなアイコンシステムの構築や実行の支援が本システムの目的である。本システムの構築環境の構成を図1に示す。

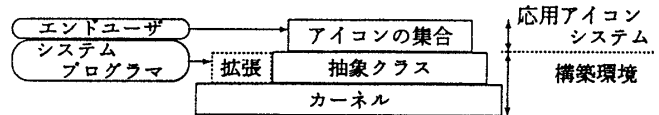


図1: アイコンシステム構築環境の構成階層

アイコンシステムはカーネルと有限個の抽象クラスと任意の個数のアイコンからなる。アイコンはシステムプログラマが与えた抽象クラスから作成され、ユーザの必要に応じて作業空間に新たなアイコンを追加することができる。抽象クラスはデータの型とその型に伴うメソッドをもつモデルである。カーネルはいくつかのマネージャからなりアイコンや抽象クラスを管理し、ユーザに所望のアイコンプログラミング環境を提供する。

本システムはユーザをエンドユーザとシステムプログラマの2つのカテゴリに分ける。エンドユーザはシステムプログラマが提供する応用アイコンシステムを用いる。エンドユーザにはアイコンシステム上でアイコンの作成、削除、操作等の機能がカーネルにより提供されている。

アイコンの作成はシステムプログラマが提供する抽象クラスのデータ型に適切なデフォルト値を与え、次にユーザや他のモジュールによってカスタマイズされることによって作成される。またアイコンは必要に応じて削除することができる。

アイコンのもつデータの操作は、ユーザによる2つのアイコンの重ね合わせによって行われる。重ね合わせが起こると互いのメッセージを交換比較し、実行可能なメソッドのリストを作成する。ユーザはそのリストの中から任意のメッセージを視覚的に選択すること

によって、アイコンのメソッドの実行をクリックエーストする。データの操作の結果は新たなアイコンとして作成されシステムに登録される。このような操作を繰り返すことによってプログラムを行う。またアイコンプログラミングの支援として一連の手続きを一つのアイコンとしてマクロ定義することができる[4]。マクロ定義されたアイコンはプログラム部品の再利用を促し生産性を向上させる。

一方システムプログラマは抽象クラスを作成してアイコンシステムを拡張することができる。抽象クラスとカーネルはメッセージを交換することによって処理の要求を行うため、システムプログラマはある一定のインターフェイスをもつクラスをアイコンシステムの抽象クラスとして定義することができる。また抽象クラスはクラス階層もっており、システムプログラマの生産性の向上に寄与している。

3 システム構成

図2に本システムの構成を示す。本システムはMain Moduleと4つのマネージャ:Window Manager, History Manager, Icon Manager, Icon-base Managerより構成され、実行時にモジュール同士がメッセージをやりとりしながら処理を行う。

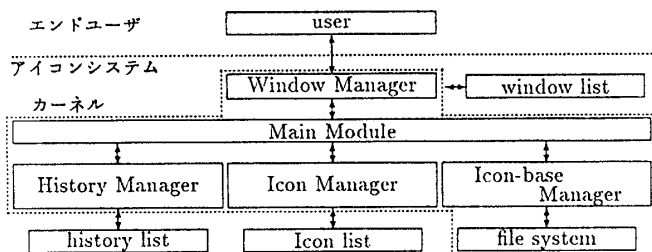


図2: 本システムのモジュール構成

1. Main Module

Main Module は各マネージャとのメッセージ交換によってシステムの動作を制御するモジュールである。Main Module の機能は、アイコンの重なり解釈、実行、システムの開始及び終了時の処理、ユーザへの各種サービスの制御を行う。

2. Icon Manager

Icon Manager はアイコンの内部データを管理する。Icon Manager への処理はユーザや他のモジュールからアイコンの作成、削除、メソッド実行やアイコンに関する各種情報を取得する際に要求される。

3. Icon-base Manager

Icon-base Manager は file system 上に存

在するアイコンのデータベースを管理するモジュールである。アイコンシステムのユーザ環境とデータを保守する。

4. History Manager

History Manager はアイコン操作により選択実行されたメッセージ等を history list として保持する。このリストを用いてマクロ定義や undo 機能をユーザに提供する。

5. Window Manager

Window Manager はユーザとのインターフェイスを担当し、ユーザの入力イベントをアイコンの外部データを管理する window list を参照して、適切な内部表現に変換する。またユーザの操作に対する結果を視覚的にユーザに報告する。

4 おわりに

本稿ではアイコンシステムの構築と実行を支援するシステムについて報告した。試作システムの開発を通して、エンドユーザに対しては有効なプログラミング環境を提供できる見通しを得た。しかし抽象クラスを作成するシステムプログラマに対しての支援ツールは設計の段階である。

現在カーネルは著者らが設計した動作しか行わないが、今後の課題としてアイコンシステムの応用分野に応じて、システムのカーネルの動作の制御を記述できるようにする考えである。これによってさらに応用アイコンシステムの構築に柔軟性をもたせることができる。

本システムは様々な課題やユーザに利用できるように設計されている。アイコンシステムはユーザの問題解決を容易に導くような環境を提供するために、アイコンの集合から課題や状況に応じて必要なアイコンを選択的にユーザに提供する機能が必要である。

参考文献

- [1] R.B.Grafton et al.(eds.), Special Issue on Visual Programming, *IEEE Computer*, vol.18, No.8, 1985.
- [2] T.Ichikawa et al.(eds.), Special Section on Visual Programming, *IEEE Tr. SE*, vol.16, No.10, 1990.
- [3] M.Hirakawa et al., "A Framework for Construction of Icon Systems", *Proc., IEEE Workshop on Visual Languages*, pp.70-77, 1988.
- [4] M.Hirakawa et al., "HI-VISUAL Iconic Programming", *Proc., Workshop on Visual Languages*, pp.305-314, 1987.