

メディア操作言語「紋様」 - 実行方式の概要 -

2M-12

小林 り恵、 小泉 忍、 山野 紘一

(株)日立製作所 システム開発研究所

1. はじめに

帳票、文書、図形、画像、音声等のマルチメディアを統一的に取扱うことを目的とした、オブジェクト指向のメディア操作言語「紋様」¹⁾の処理系の実行方式について述べる。

「紋様」言語は、

- (1) テンプレートによるデータ抽象化機能、メソッドの借用による継承機能を実現した、オブジェクト指向機能
- (2) インスタンスの外部表現仕様であるメディア型機能
- (3) 複数のインスタンスの組み合わせによるプログラミングを支援するためのオブジェクト制約機能とオブジェクトリンク機能
- (4) データ型、関数定義等の機能を持つ。

「紋様」処理系は、これらの複数の機能を統一して効率良く処理することが必要であり、その実行方式を考案した。以下、「紋様」処理系の動作環境、実行方式の概要について述べる。

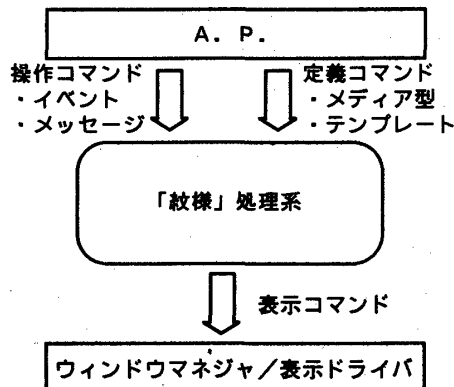
2. 「紋様」処理系の動作環境

「紋様」処理系の動作環境を第1図に示す。

アプリケーションプログラム全体を紋様で記述するのではなく、その中のマルチメディア処理部分のみに紋様を利用すること、および、対話処理によるテンプレートやインスタンスの動的編集(生成、更新、リンク等)に対応することを想定し、独立した実行系として「紋様」の処理を行なう。

3. 実行方式の概要

「紋様」の機能を大別すると、定義機能(テンプレート、メディア型、関数の定義)と操作機能(インスタンス生成、メッセージ通信、表示等)に分類

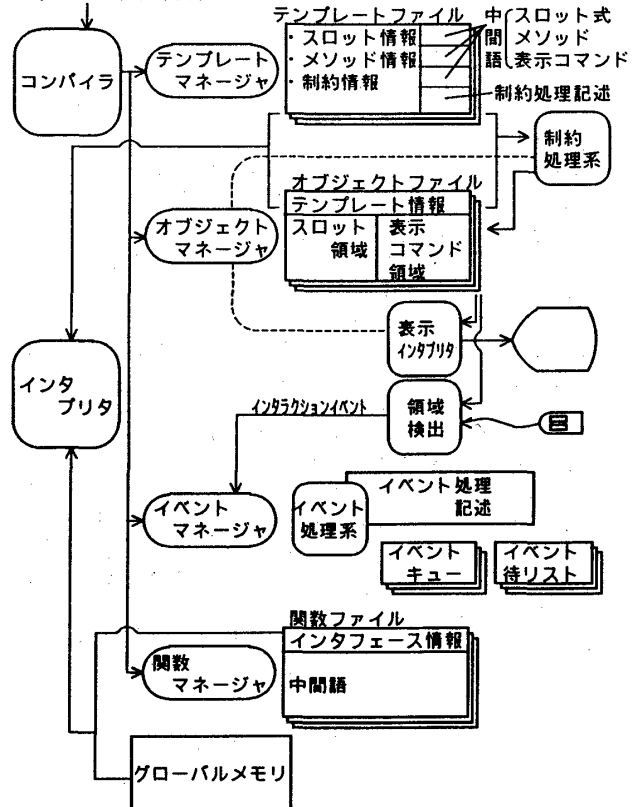


第1図 「紋様」処理系の動作環境

できる。前者による定義内容は、例えばテンプレート定義のように実行中に変更がなく、かつ、インスタンス生成のような操作にもなって複数回参照される。これらの実行の高速化と移行性・保守性の両立を狙って、仮想マシンを設定し、処理のトップレベルにおける1コマンド毎のコンパイル/即実行という方式で上記インタプリタを実現している。紋様処理系の概要を第2図に示す。なお、以下では、紋様プログラムを仮想マシン語に変換する部分をコンパイラ、変換後の仮想マシン語を解釈実行する部分をインタプリタと呼ぶ。

3.1 実行方式

コンパイラは、「紋様」で記述されたプログラムやコマンドの構文解析を行ない、仮想マシン語に変換した後、解析した定義やコマンドの、それぞれの内容に従って、対応するマネージャを呼ぶ。各マネージャは、それぞれの定義および機能の実行に対応



第2図 「紋様」処理系の概要

る処理を行なう。

その他、制約機能に対応する制約処理系、表示コマンドを実行する表示インタプリタ、キーボード/マウス等やオブジェクトによるイベントの起動に対応するイベント処理系などがある。

(1) テンプレートマネージャ

テンプレートマネージャは、テンプレート定義およびメディア型定義に対応して、テンプレートファイルを作成管理する。テンプレートファイルには、インスタンスへのアクセスや値の設定に必要なすべての情報(下記)を設定し、他のテンプレートのコンパイルやコマンドの実行時に参照する。

- i) スロット情報: スロット名、データ型、オブジェクトレコード(後述)内のアドレス(オフセット)およびメモリサイズ、スロットに個別に与えられたスロット式を実行する仮想マシン語列。
- ii) メソッド情報: メソッド名、パラメタの仕様(パラメタ名、データ型、メモリオフセット)、メソッドを実行する仮想マシン語列/メディア型の場合オブジェクトレコード上に表示コマンドを作成する仮想マシン語列。
- iii) 制約情報: 制約対象のスロットのリスト、制約式の間置語。

(2) オブジェクトマネージャ

オブジェクトマネージャは、インスタンスの内部データ構造であるオブジェクトレコードの生成・消滅・アクセス管理を行なう。オブジェクトレコードの集まり全体をオブジェクトファイルと呼ぶ。オブジェクトレコードは、管理領域、スロット領域、表示コマンド領域からなる。

管理領域には、オブジェクトマネージャが、テンプレートID、インスタンスID、およびメモリ管理ポインタを設定する。スロット領域とは、スロットの値を実行中に保持する領域である。また、表示コマンド領域とは、メディア型に対応した表示コマンドを展開する領域である。

この表示コマンドを表示インタプリタが実行することにより、当該インスタンスがディスプレイ上に表示される。また、各インスタンスに対するマウスによるピッキングも、この表示コマンドに基づいて判定する。スロット領域および表示コマンド領域にはスロット式やメソッドに対応する仮想マシン語の実行および制約処理によって値が設定される。

(3) イベントマネージャ

イベントマネージャは、ユーザとの対話処理(キーボード/マウス入力とそれに対する応答処理)であるイベント処理の管理を行なう。イベント処理には、イベントの発生元(入力元)の名称と受け渡すパラメタを定義するイベント宣言、ユーザやオブジェクトによって、キーボード、マウス等による入力が行なわれるイベント発生、オブジェクトにおいて指定されたイベントの発生を待つ(入力待ち)イベント待ち等がある。イベント宣言に対しては、イベント処理記述を生成し、イベント発生に対しては、イベントキュー、イベント待ちに対しては、イベント待ちリストを生成する。

(4) 関数マネージャ

関数マネージャは、関数定義に対して、その関数と外部とのインタフェース情報、関数の処理内容のマシン語情報を記憶した関数ファイルを作成する。

関数定義では、整数型、実数型、論理型、文字列、ビット型、文字列型、ビット列型、列挙型、直積型、

リスト型、ベクトル型、マトリックス型のデータが扱え、整数/実数の四則演算や比較演算、文字列やビット列の連結や要素の追加などの演算を提供している。さらに、関数マネージャは、算術関数やリスト操作の処理等の組み込み関数を提供する。関数呼出し時には、対応する関数の関数ファイルと、グローバルな変数の領域のアドレスとをインタプリタに渡して実行させる。

(5) 制約処理系

制約処理系には、スロット式による値の再計算と、制約式による制約の解消の2つの機能がある。

スロット式による制約では、スロットの値が常に式の値に一致するように再計算する。

制約式による制約の解消では、制約式をそのまま記憶するために木構造の間置語を採用している。制約式は、正規化された式の間置語に変換し制約解消を実行する。間置語への変換は、定数伝播、線形方程式の解決、非線形方程式の線形化の3つを適宜繰り返すことで行なう。この計算は、有理数領域で行なう。

3. 2 仮想マシン語とデータ形式

仮想マシンは、演算用スタックを持ち、四則演算をこのスタック上で行なう、スタックマシン型の実行方式を採用している。

仮想マシン語には、固定長データに対する四則演算/比較演算等の演算命令、可変長データに対する生成/追加/結合/比較等の操作関数がある。

仮想マシンで扱うデータ型は、整数型や実数型等の固定長データ、文字列型やリスト型の可変長データ、およびこれらを要素とする構造型データの3種である。

固定長データと構造型データは、ベースアドレスと固定オフセットを用いたアドレッシング方法によりアクセスの高速化を図っている。可変長データへのアクセスは、各データに固有のデータ情報テーブルを介して間接的に行なう。

可変長データは、データ変更時には新たなデータ領域を作成するので、変更前のデータ領域は無効となる。また、関数等にローカルなデータのデータ領域は、その関数の実行が終了しても存在しているので、これも無効領域となる。そこで、これらの無効領域の増大によって起こる領域不足の解消のために、ガーベジコレクションを行なっている。

さらに、仮想マシンでは、操作表示インタプリタや実際のオブジェクトを管理しているオブジェクトマネージャ等の仮想マシン語プログラムを実行する、汎用形の外部プログラム呼出し用機能も備えている。

4. おわりに

帳票、文書、図形、画像、音声等を取扱うことを主眼とした、オブジェクト指向のメディア操作言語「紋様」の処理系の実行方式について述べた。この実行方式により、オブジェクト指向、メディア型、オブジェクト制約、オブジェクトリンク、関数定義等の複数の言語機能を、統一した実行方式で効率良く処理することができる。

<参考文献>

- 1) 小泉, 小林, 山野:
メディア操作言語「紋様」—言語の特徴—
情報処理学会第42回全国大会