2M−11

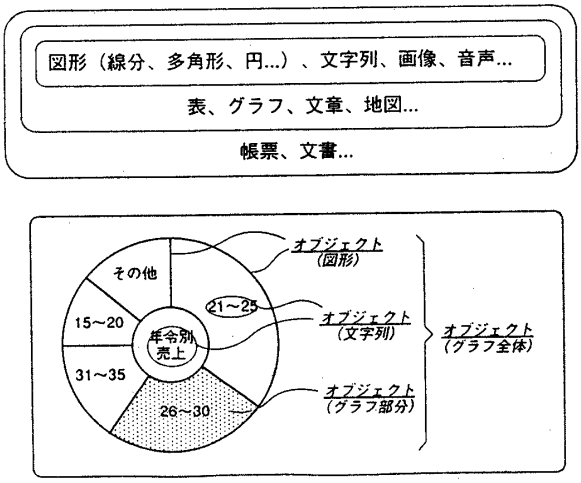
メディア操作言語「紋様」 −言語の特徴−

小泉　忍、　小林　り恵、　山野　紘一

（株）日立製作所　システム開発研究所

1．はじめに

マルチメディア処理では、対象がバラエティに富んでいるだけでなく、それぞれの操作自身も複雑である。このため、従来の数値データ処理等で一般的なサブルーチンパッケージでは、サブルーチンの数が多く、かつ対象と適合するサブルーチンの組み合わせやパラメータの適合性の検証に関して、利用者（プログラマ）に大きな負担を強いることが問題である。このような問題に対しては「オブジェクト指向」アプローチが有効である。メディア操作言語「**紋様**(**Monyoh**:**M**edia **O**peration Language System for Humanware)」は、帳票、文書、図形、画像、音声等を処理の対象＝「オブジェクト」としてとらえ、それらを統一的に取扱うことを主眼とした言語である。特に、図形、音声、画像等の要素からなる、文書、帳票等の複合オブジェクト(**第１図**)を取扱うための「オブジェクトリンク機能」「オブジェクト制約機能」を特徴としている。



第１図　紋様の対象世界

2．言語の特徴

紋様の主要な機能を**第２図**に示す。

2．1　オブジェクト指向機能

オブジェクト指向言語の一般形は、「データ抽象化機能」＋「継承機能」である。紋様では、これらを以下のように取扱っている。

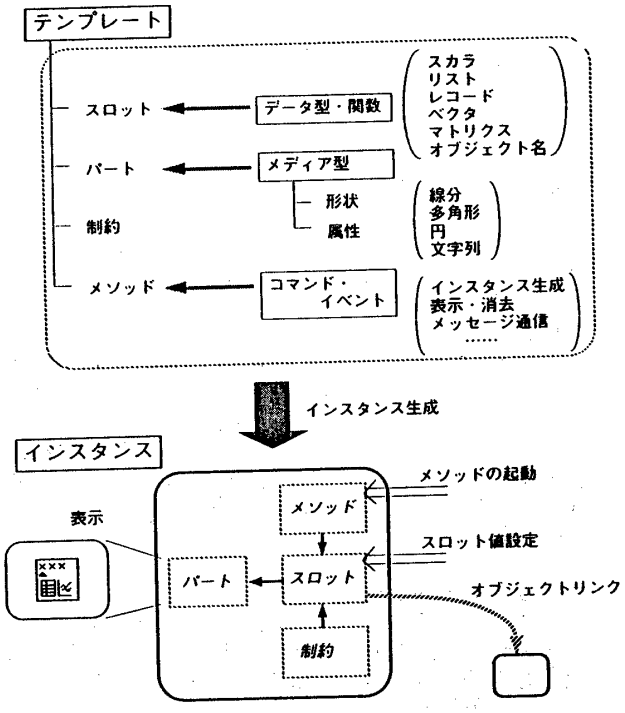
（1）データ抽象化機能

紋様では、「テンプレート」によりインスタンスの構造と操作の仕様を定義し、データ抽象化機能を実現している。ここで、インスタンスの構造とは、スロット（インスタンスの状態変数）の構成、および、表現仕様（⇒2．2メディア型）の指定である。また、操作の仕様とは、メソッド（スロットの更新、および他のメソッドやコマンドの適用手順）の定義である。テンプレートは、一般のオブジェクト指向言語における「クラス」に相当するが、紋様では、テンプレート自身はインスタンスの範型を指定するものであってオブジェクトとして取扱わない。

なお、インスタンスの生成は、定義済のテンプレートに基づいて、生成コマンドにより行なう。インスタンスがオブジェクトとして取扱われる。

（2）継承機能

紋様における継承機能は、メソッドの借用(delegation)により実現している。継承機能の目的の一つは「差分プログラミング」を実現することであり、一般のオブジェクト指向言語では、継承は「クラス階層」に基づき、親クラスからの変更点のみを再定義する方法をとっている。一方、紋様では、定義済みの任意のテンプレートのメソッドを新たに定義するテンプレートで引用することにより、メソッドの流用を可能としている。



第２図　紋様の概要

Media Operation Language "Monyoh" - Language feature -
Shinobu Koizumi, Rie Kobayashi, Koichi Yamano
Systems Development Laboratory, Hitachi Ltd.

2．2　メディア型機能

メディア型とは、インスタンスの外部表現仕様である。即ち、図形（線分、円、多角形、文字列...）画像、音声等の識別、それらの表現時の属性（線種、色、大きさ、位置等）、およびそれらの組み合わせである。紋様では、既定のメディア型としてグラフィックプリミティブを提供しているが、それらを組み合わせた新たなメディア型の定義が可能である。

なお、表示等の外部表現を制御するものとして、表示、消去の各コマンドを設けている。

2．3　オブジェクト制約とオブジェクトリンク

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、紋様では「オブジェクト制約機能」と「オブジェクトリンク機能」を設けている。

（1）オブジェクト制約機能

メソッドによらずにスロットの値を計算するメカニズムとして、紋様には、以下の2つのオブジェクト制約機能を設けている。

a）個々のスロットに付与された「スロット式」による制約

b）テンプレートに対して付与された「制約式」（＝複数スロット間の値の関係式）による制約

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式が再評価され、その値がスロットに代入される。また、制約式による制約では、制約式が成立するように、値が明示的に与えられていないスロットの値が調整される。いずれの場合にも、スロットの値の決定のタイミングは、プログラム上で直接指示するのではなく、処理系が必要に応じて計算を実施する。

なお制約機能を持つオブジェクト指向言語には、ThingLab II[1] (Washington大)、Coral[2] (CMU) 等があるが、いずれもユーザインタフェースの分野に特化している。

（2）オブジェクトリンク

オブジェクトリンクとは、あるインスタンスから他のインスタンスのスロットの値を（メッセージ通信等を用いずに）直接的に参照する機能である。このため、参照先の他のインスタンス名を保持するための「オブジェクトスロット」と、オブジェクトスロットを介して、参照先のインスタンスのスロットを間接参照する機能を設けている。

上記オブジェクト制約とオブジェクトリンクとを併用することにより、依存関係のある複数のインスタンスを動的に結合することが簡単に実現できる。**（第3図）**

2．3　その他の機能

紋様におけるその他の機能には、データ型、関数定義等がある。これによって、算術演算、リスト処理、および座標変換に必要なベクタ・マトリクス演算等を実現している。

3．言語の設計思想

紋様の言語設計に当たっては、紋様を用いたオブジェクトライブラリ（テンプレート群）の作成とアプリケーションによるオブジェクトライブラリの利用（インスタンスの生成／組み合わせ複合化／操作）との2つの利用フェーズを想定した。

ライブラリ利用者の立場からは「モジュラリティの高さ」が要求される。モジュラリティには、要素の独立性とそれらの結合性の2つの側面がある。紋様では、前者に対しては、テンプレートによるデータ抽象化機能と内部状態を自律的に決定する制約機能により、また、後者に対しては、オブジェクトリンク機能と外部状態に追随する制約機能により、対応を図っている。

一方、ライブラリ作成者の立場からは安全性よりも「自由度の高さ」や「実行性能の高さ」が要求される。これは、テンプレートの定義方法や定義内容に反映させた。例えば、継承の方法として、制限の多いクラス階層を用いずに借用を採用したこと、他のオブジェクトのスロットの値を参照する方法として、メッセージ通信によらない直接的な参照を可能にしたこと等である。

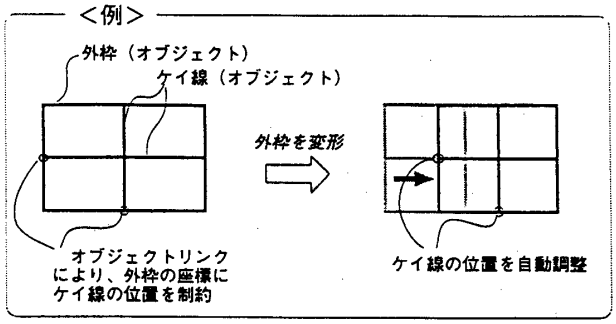
4．終わりに

帳票、文書、図形、画像、音声等を取扱うことを主眼とし、処理対象を部品化しそれらを有機的に結合することによるプログラミングを支援する「オブジェクトリンク機能」「オブジェクト制約機能」を特徴としたメディア操作言語「紋様」について述べた。

＜参考文献＞

1）Maloney, John H.; Borning, Alan; Freeman-Benson, Bjorn N.: "Constraint Technology for User-Interface Construction in ThingLabII", Proceedings of the OOPSLA'89 (Oct. 1989) pp.381-388

2）Szekeiy, Pedro A.; Myers, Brad A.: "A User Interface Toolkit Based on Graphical Objects and Constraint", Proceedings of the OOPSLA'88 (Sept. 1988) pp.36-45



第3図　オブジェクト制約／オブジェクトリンク