

3K-6

多種データ形態適用型 エディタの実現方式

小笠原 達男^{*}, 渡辺 豊英^{*}, 吉田 雄二^{*}, 福村 晃夫^{**}

(*名古屋大学 工学部)

(**中京大学)

1. はじめに

編集機能は計算機システムの最も基本的な機能として発展し、数多くのエディタが開発された。しかし、計算機利用の拡大とともに、編集対象も英数字データに加えて、漢字データ、図形データ、画像データなどにおいて、多様な形態・構造のデータを応用目的に応じて編集することが重要な課題となっている。今日までに開発された多くのエディタは、応用用途に強く依存した編集機構を採用し、統一的な操作インターフェースの下に多種多様なデータを編集可能とする枠組みで設計されてない。本稿では、統一的な操作インターフェースの下で、多様な形態・構造のデータを処理可能にする編集機構の枠組みと実現方法について報告する。

2. データ形態・構造に依存しない編集機構

様々な形態・構造のデータを統一的な枠組みの下で扱うには、編集機構が特定の処理データに従属しないことが必要である。すなわち、文字データ、数値データ、図形（ベクトル）データ及び画像データなどの物理的に性質が異なったデータ・オブジェクトを統一的に扱うには、データベースの構成概念が有用である。データベースの枠組みでは、データ構造を物理構造と論理構造に分けて捉え、システムが処理する物理構造とユーザが操作する論理構造の構成概念は、物理構造に対して様々な論理構造を提供する。この概念に基づいて、物理的に性質が異なったデータ・オブジェクトを、データ定義情報で指示されたスキーマの下に統一的に操作可能とする。図1に、この操作概念を示した。

3. データ定義言語と編集操作コマンド

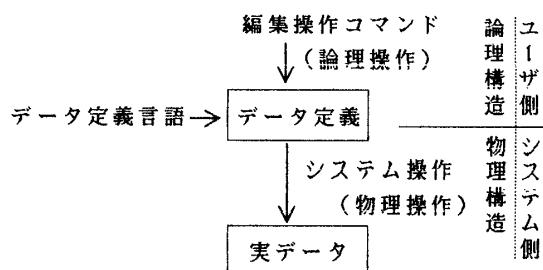


図1 論理構造と物理構造の操作概念

論理的な編集データの形態・構造を記述するために、データ定義言語を用いる。リレーションナル・データベースにおけるデータベース定義言語とほぼ同じ記述形式を採用し、編集データを複数の属性からなるレコードの集合と考える。例えば、論文形式の論理構造を定義すると、以下のようにになる。

[例1]

```
structure PAPER;
  term TI      : char(50);
  term AU(5)   : char(20);
  term AR      : char(10000);
  term PIC     : vector(100);
end;
```

個々の構成要素に独立なデータ属性を指示し、全体として一つのテーブルを定義する。

更に、本データ定義言語では、データの特性に応じた固有の機能（以下、付加手続きと呼ぶ）を構成要素に指示可能として、ユーザの編集処理操作を支援する。このような機能の一つとして、プログラム編集時の文法チェック機能があり、以下のようにデータ定義言語でデータ・オブジェクトの属性に対して定義する。

[例2]

```
structure PROGRAM;
  proc syntax-check(FORT77);
  term PR(1000) : char(80);
end;
```

A system configuration of the multimedia-oriented editor

Tatsuo OGASAWARA*, Toyohide WATANABE*, Yuuji YOSHIDA*, Teruo FUKUMURA**

*Faculty of Engineering, Nagoya University, **Chukyo University

編集操作コマンドは、以下のような一般的な構文を有する。これは、コマンドやデータ・オブジェクトによらず、一定であり、コマンドによって、またデータ・オブジェクトによって<source>や<destination>の解釈が行なわれる。

[コマンド]

<command> <source> (<destination>)
<source>で指示したデータ・オブジェクトが、<destination>で指示するデータ・オブジェクトに対して、<command>が表わす処理を実行する。<source>及び<destination>は、データ・オブジェクトの名前や、名前に付帯した行番号や領域を表わす。例えば、

[コマンド記述例]

```
LIST ARTICLE
COPY ARTICLE1:10-20 ARTICLE2
```

は、それぞれARTICLEが表わすデータ・オブジェクトの表示、ARTICLE1が指示するデータ・オブジェクトの10行目から20行目までをARTICLE2が指示するデータ・オブジェクトへの複写を指令する。

4. システム構成

本編集機構では、単純に操作コマンドを実行できない。操作コマンドの実行には、コマンドの入力を受けて、データ定義情報を参考し、パラメータが指示するデータ・オブジェクトを調べ、それがどの実データと対応するかを決定して実操作する。

データ定義情報や操作コマンドの内容によっては、処理途中でデータ定義情報と実データの対応関係を調べたり、また付加手続きに対してのインターフェースを設定する必要がある。本編集システムは概略図2に示した構成となり、6つの主要なモジュールから成る。

- (1) コマンド入力
- (2) マッピング管理
- (3) 編集処理
- (4) データ定義情報管理
- (5) 実データ管理
- (6) 付加手続き管理

それぞれのモジュールを簡単に説明する。コマンド入力部は、ユーザとシステムのインターフェース部分である。ユーザが入力した操作コマンド及びパラメータを編集処理部に送り、処理の起動を指令する。マッピング管理部では、デー

タ・オブジェクトを文字、図形、画像などの属性に合わせて、ディスプレイ画面に表示したり、表示されているデータ・オブジェクトを管理する。編集処理部は、コマンド入力部からの操作コマンドやパラメータを受理して処理する。また、データ定義情報管理部や実データ管理部と連携してデータ・オブジェクトの属性や実データを取り込み、操作コマンドで指示された処理をデータ・オブジェクトの属性に一致するように制御する。さらに、付加手続きが指示されていれば、それを呼び出す。データ定義情報管理部は、データ定義言語で指示された定義名とその属性、実データの対応関係などを一括管理する。定義名(termの項目名)による参照要求には、その属性と対応する実データを調べる。実データ管理部は、バッファ管理のように働き、実データをファイルからメモリ上に取り込む。またメモリに空き領域がなくなったら、メモリ上のデータをファイルに書き出して空き領域を確保する。付加手続き管理部は、編集処理に付帯する手続きに対してインターフェース結合を図る。付加手続きはシステムに予め登録されており、パラメータの引渡しとともに呼び出される。

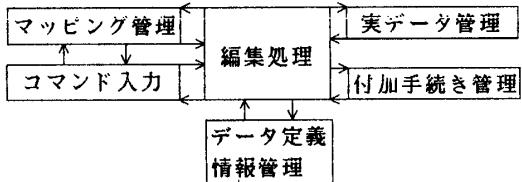


図2 編集システムの概略

5. おわりに

本稿では、多種のデータ形態に対して統一的な操作で編集可能とする機構を示し、この機構を実現するシステムの機構を説明した。そして、編集機構がデータ形態・構造に依存しないと同様に、システム構成も各モジュールが極力独立に構成できることを報告した。現在、開発中であり、細部の構造については今後報告する予定である。

〔参考文献〕

- 1) 渡辺他：「編集データの形態・構造に依存しないエディタの機構」、情報処理学会第36回（昭和63年前期）全国大会講演論文集、pp885-886。