

分散システムにおけるファイル構造の提案

龍 忠光 寛 元 青江 茂

富士通ネットワークエンジニアリング㈱ (FNE)

現在、ユーザの保有している分散情報ネットワークにおける戦略情報システム(SIS)の構築が強く要望されるようになった。本稿では情報表現モデルとしてオブジェクトセンサー方式を提案し、情報表現モデルの階層化について報告する。これにより、ネットワークやターミナルの存在を意識せずマルチメディアデータが取り扱え、“ハイパー言語”の採用によりユーザによって目的に合ったデータリンクや入出力データのフォーマットを自由に作る事が可能となる。

A PROPOSAL OF FILE STRUCTURE FOR
DISTRIBUTED SYSTEMS
Tadamitsu Ryu , Gen Kakehi , Sigeru Aoe
FUJITSU NETWORK ENGINEERING LIMITED
KANAGAWA SCIENCE PARK, 100-1 SAKATO
TAKATSU-KU KAWASAKI-CITY , 213 , JAPAN

There are increasingly strong needs for Strategic Information Systems in distributed systems.

In this paper we propose the "object-sensor system" in information model and the hierarchical information model by which we can construct user-customized file systems.

By these architectures user can handle multimedia data easily and can make several types of data links and input-output data formats using hyper-language.

1. はじめに

ネットワークにおいては電話回線から高速デジタル回線が出現し、ISDN、B-ISDN、SDN(Software Defined Network)を基調としたインテリジェントネットワーク(IN)又、衛星回線や無線LANさらに共通線信号方式などにおけるブロードキャスト的コントロール方法も提供されつつある。INについてはそのサービスの追加はシステムを止めることなくオンラインで行うなどハードウェアへの高度な要求もされることながらソフトウェアの高度な処理が要請されている。分散されておかれたターミナルからネットワークのノードのコントロール、さらには相手ターミナルのソフトウェアアクセスさえ要求されている。

以上のような高度な環境のもとでユーザは戦略情報システム(SIS)の構築を強く要求するようになって来た。その中でスペース的に広がったユーザの情報ネットワークを使った分散システムの構築は避けて通ることができない。ユーザは

- ①ターミナルの追加、メモリの追加ができるフレキシブルな設計
- ②容易にサービス追加や変更のできるシステム
- ③ニーズに合わせた容易なデータベースの構築などを要求している。

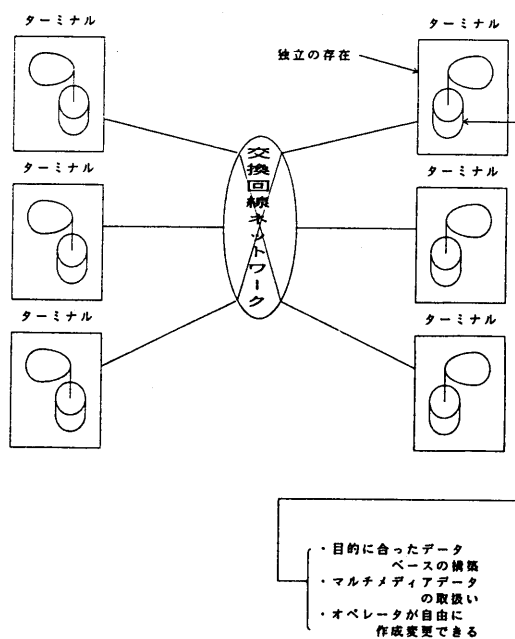
この要望を実現するために分散システムにおけるデータのやりとりをプログラムのやりとりまで包含して考察し、それらを管理する手法について検討した結果を報告する。

2. 分散システムとは

分散システムといったとき、対象としている条件とはなにかを述べてみる。もともと分散システムと集中システムの違いはデータへのアクセスの度合いによる。オペレータが必要とするデータの多くが、分散されている自分のターミナルにある程度集中しているからこそ、各オペレータのターミナルを分散して設置する意味がある。

さらに分散システムにおけるネットワークの動きは現在スペース的に狭い領域におけるLANか

ら広い地域におけるISDN、そしてB-ISDNにより扱える信号が狭い帯域から広い帯域へと広がり、デジタル技術の進歩がメディアの違いの壁を取り除く。SDNの出現によって交換回線をあたかも専用回線のようにサービスする時代へと様変わりしようとしている。共通線信号方式によって接続する時間もダイヤル方式の10数秒から10数ミリ秒と短縮される。このネットワーク環境は現在のローカルLANの分散システムの世界からダイナミックなグローバルネットワークの世界ができてきたことを意味する。以上の条件のもと、ここでは以下のような分散システムを対象として考える。(図1)



(図1) 分散システムの概念図

- ・各ターミナルを結びつけているネットワークは交換回数で不特定多数の人と接続できコミュニケーションができる。
- ・各ターミナルはすべて同格であり、独立である。
- ・そのため各ターミナルは1つのシステムとして完結している。すなわち各ターミナルは本来がコンピュータシステムであり、データベ

ースシステムである。

- ・そして他ターミナルとの間に分散処理があり、データのやりとりができる。
- ・各ターミナルの能力（容量、性能、機能など）は同一でなくてもよい。
- ・分散システムのデータ変更時の同時一致処理を取らなくてもシステムティックに処理できる。

さらに、分散システムを使用する側すなわち、ヒューマンインターフェース的に考えたときはどのような条件があれば望ましいかを以下に述べる。

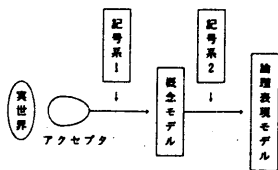
- ・ネットワークやターミナルの存在と分散を意識させないで操作できる。
- ・各ターミナルには処理の目的に合った各種リンクが組める。
- ・マルチメディアが取り扱える。
- ・オペレータによって入力及び出力フォーマットが自由に決められ、変更、追加ができる。
- ・各ターミナルで構築されたデータベースを自由に結びつけられ検索できる。

3. 情報表現モデルの仮想化

データすなわち情報をマルチメディアデータにモデル化し、それをリモートで取扱い、自由にリンク付けし、オペレータにシステムの物理的存在を意識させないで分散システムを動かす方法について述べる。

3-1) 情報表現モデル

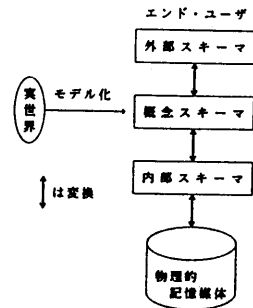
実世界のマルチメディアデータのモデリングの概念は、増永 [8] によって既に報告されている。それによると図2に示すように、データベース構築は2段階で行われる。第1段階で



(図2) 実世界のデータモデリング

は実世界を眺むアクセプタによって性質も情報量もまったく異なるメディアデータが“概念モデル”に変換される。第2段階では概念モデルをコンピュータ上で実働可能な表現にモデル変換され“論理表現モデル”化される。これにより、各種メディアデータのまちまちの大きさが1つの記号で表されることになり、統一されて取り扱われることになる。さらにデータベースの性質上、ソフトウェアからの高い独立性、すなわち物理データの独立性、論理的データの独立性を保つために、図3に示す三層スキーマ構造が採られている。

データを分散すなわちリモートで取り扱うためには、各ターミナル間でのやりとりはなるべく転送時のオーバーヘッドを少なくするために、一定の大きさで、容量の小さいデータが望まれる。このデータは、上記で論じた概念スキーマ及び外部スキーマに当てはめると分散システムにとって親和性があり都合がよい。



(図3) ANSI/×3/SPARC のDBMSの三層スキーマ構造

リモートでやりとりされるデータは必ずその属性が定義付けられており、そのデータを受け取ったターミナルにおいてもこの属性にあわせた処理をすることになる。ここにリモートでやりとりするデータにその属性をも分かるように記号化し、コマンド化する情報表現モデルを定義する。このコマンドコードは受け取ったターミナルで自分の振るまいを知っていることになり、そこで必要なプログラムを選択し動くことができる。受け取ったターミナルに必要なプロ

グラムがなければそのプログラムをコマンドコードを送り込んだターミナルから学習（ラーニング）することを義務付けておけばこのコマンドによって情報表現モデルはアクターモデルと考えられる。このコマンドのことをオブジェクトセンサーと呼ぶことにする。オブジェクトセンサーは論理表現モデルと物理的記憶媒体に入っているエンティティ（概念スキーマ）を表している。前者をオブジェクトコマンド（以下コマンドと略す）と呼ぶ。

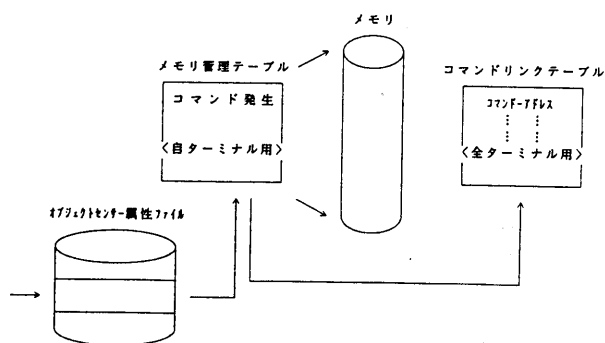
3-2) オブジェクトセンサーの属性コンセプト

オブジェクトセンサーは実世界の概念モデルと論理表現モデルの両モデルを表した呼び名である。ここでのオブジェクトセンサーにその性質を示す属性を与える考えを導入する。我々はこのオブジェクトセンサーの属性を与えることを“登録”と呼んでいる。

オブジェクトセンサーを登録するということはそのオブジェクトセンサーのエンティティデータに目を向けているのではなく、そのエンティティデータの入れ物（概念スキーマ）に注目してその入れものの属性を決めていることになる。その入れものの大きさすらその入れものの1つの属性にすぎないため、部品の大きさはオペレータの思いのままに指定することができる。一旦決定した属性もいつでも、どこでも自由に変えられる。大きさは自由であり、その中に扱われるデータの性質も自由に決めることができることから大きさも性質もマルチメディアデータを一律に取り扱うことができる。その入れものにコードを与え、その入れものの性質を表す記号をも一緒につけることでその入れものの名前をみるだけでそのインスタンスがどのメソッドを指定しているのかも分かる。このオブジェクトセンサーのコマンドを使って処理する方法は基本的にスピードアップとメモリ減に役立つ。具体的な属性とは名称、略称、コメント、キーワード、コマンド名、格納場所、……などである。これらの属性を管理するファイルを属性フ

ァイルと呼びシステムで使用するすべてのオブジェクトセンサーはこの属性ファイルに登録することで意味付けされる。この属性も1つのオブジェクトセンサーである。

図4にオブジェクトセンサーの登録時のコマンドコード発生とコマンドリンク生成の原理を示す。オブジェクトセンサー登録によりコマンドが内部コードとして割り当てられ、そのコマンドに対するリンクがとられる。前者がメモリ管理オブジェクトの働きであり、後者がコマンドリンクオブジェクトの働きである。図5にメモリ管理オブジェクトとコマンドリンクオブジェクトの働きを示す。メモリ管理テーブルをインスタンスで持つメモリ管理オブジェクトは基本的には自分のターミナルを対象にコントロールするオブジェクトであり、自分のターミナルを監視管理する。オブジェクトセンサーが登録されるとコマンドコードをその属性に従って割り付け、そのコマンドコードのエンティティデータの入るエリアを直接割り付けてくれる。



(図4) オブジェクトセンサー登録時の動き

このため、メモリ管理オブジェクトは自分のターミナルの空きエリアの管理、エリアの割り当て管理を行う。エリアの大きさはオペレータが指定する場合もあるし入力データによって自動的に指定される場合もある。さらにこのメモリ管理オブジェクトは自分のメモリが不足したり

オペレータからの他ターミナルへの代行を指示されるとターミナルのメモリをあたかも自分のメモリの如く管理する役目も果たす。この場合、このメモリ管理オブジェクトは他ターミナルのメモリ管理オブジェクトと自らやりとりを行い、お互いにメモリ管理を行う。後述するがこのやりとりを同一オブジェクト・メッセージパッシングと呼ぶ。

メモリ管理オブジェクト (自ターミナルコントロール)

- (オブジェクト)コマンドコードの割り振り
- 空きエリア, 予約エリア管理
- ファイル代行管理

コマンド, リンク, オブジェクト (全ターミナルコントロール)

- スペード向上実データ変換, リンク取り
- データの可変長管理
- データのキュー管理
- データ追加・変更・削除

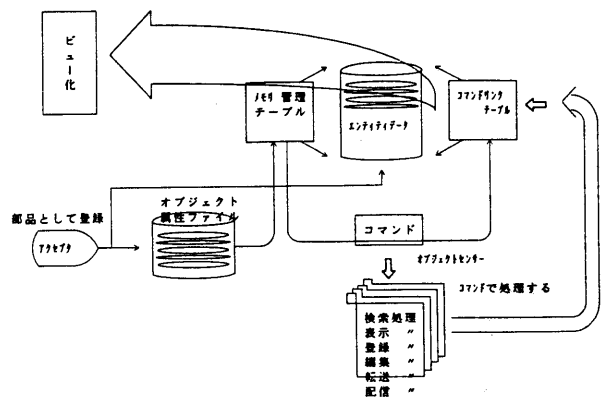
(図5) メモリ管理オブジェクトとコマンドリンクオブジェクトの働き

コマンドリンクテーブルをインスタンスで持つコマンドリンクオブジェクトは基本的には全ターミナルを対象としてコントロールするオブジェクトであり、自分のターミナルで与えられるコマンドのみではなく他ターミナルのコマンドについても管理する。メモリ管理オブジェクトによりコマンドコードとエンティティデータの入力エリアが与えられるとコマンドリンクテーブルにコマンドエリアのリンクデータが展開される。コマンドコードを示せば直ちにエンティティデータに変換する。この構造のため、エンティティデータの大きさがまちまちであっても取り扱うことができる。エリアのデータはエンティティデータの大きさを表しているからである。コマンドデータさえあればエンティティデータの頭出しが分かることから、いくら大きさの違うデータをランダムに並べてもその検

索の時間は一定で、早い。また同じコマンドにエンティティデータを1つだけでなく、多数持つこともできる。これがキュー管理の原理である。キュー管理されるデータの大きさも必ずしも同じ大きさでなくてもいい。このことからいつでもどこでもこのコマンドリンクオブジェクトを使えば、キュー管理をしかけることができる。版数という概念でデータをダイナミックに管理することもできる。さらにこのコマンドリンクテーブルにより、コマンドの追加、削除が容易に行える。追加は簡単なリンク構造で行えるし、削除はコマンドに消去ビットをたてるだけでよい。エンティティデータの変更時、エリアが不足するような場合は消去ビットを立て、新しく同じコマンド名で登録しなおせばよい。消去されたコマンドはオペレータやプログラムからの整理オブジェクトの起動により、ソートと同時に消しエンティティデータの消去も行う。

3-3) オブジェクトセンサー処理

図6にオブジェクトセンサーの処理全体の動きを示す。各種データをオブジェクトセンサーとして登録することにより、そのオブジェクトセンサーの入力位置と容量及びコマンドが割り当てられ、そのコマンドとエンティティデータの入るアドレスとサイズがコマンドリンクテーブルに展開され、その後エンティティデータが

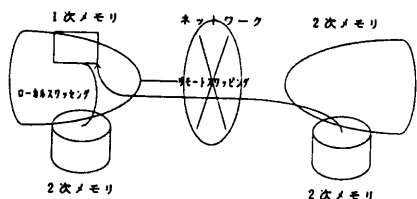


(図6) オブジェクトセンサーの処理

格納される。この一連の動きはアクセプタからのエンティティデータに概念スキーマ（コマンド）と内部スキーマ（アドレスとサイズ）が割り当てられ、そのエンティティデータが物理的記憶媒体に格納されることと一致する。このコマンドをデータとして各種応用プログラムたとえば検索処理、表示処理、登録処理、転送処理、配信処理などを行い、結果として選ばれたコマンド群はコマンドリンクテーブルによりビュー化する。これが、外部スキーマ処理であり、後述している“画面による動作組合せ処理”でその動きのシーケンスを決定する。上記の各種処理の中にリモートにおける分散処理も含まれていることはいうまでもない。

3-4) 拡張メッセージ・パッシング

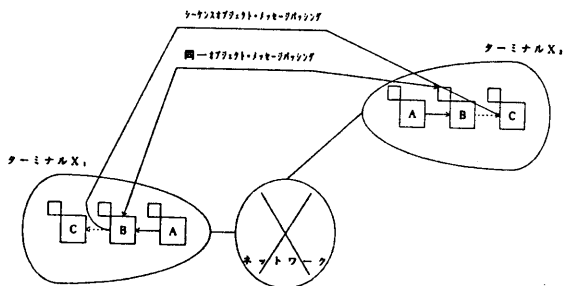
分散処理に使われるやりとりはオブジェクト指向におけるメッセージ・パッシングである。オブジェクトというプログラムモジュールは、その間をメッセージという高級な言葉でやりとりを行う。このため各オブジェクトモジュールはすべてサブルーチンのように見える。オブジェクト間のやりとりはなにも自分のターミナル内のコモンバス内だけでやるのではなく、そのコモンバスを一般のネットワークに広げて行うこともできる。図7に示すように外部すなわち分散でのオブジェクトのメッセージ・パッシングを内部、すなわちターミナル内におけるメッセージ・パッシングと同じものとみれば、自分のターミナルを一次メモリと見なし、他のターミナルすべてを二次メモリと見なす新しいスワッピング処理、すなわち分散処理が出現する。



(図7) リモートスワッピング

自分のターミナルでのスワッピングをローカルスワッピング、他のターミナルからのスワッピングをリモートスワッピングと呼んでいる。

リモートスワッピングするとき、二次メモリに記憶するようにしておけばプログラム・ダウンロードと同じものになる。従って、立上げ時に一ヶ所だけにプログラムをインストールしておけばよい。リモートスワッピングということから他のターミナルにあるプログラムを使うとTSSの機能も実現できる。メッセージ・パッシングでデータのみを渡し、他ターミナルで処理してもらえば自分のターミナルにプログラムをスワッピングしないで処理することもできる。これは従来のTSSそのものの動きと同じである。他ターミナルにあるデータをその他のターミナルのメソッドで処理したり、その逆にあるターミナルのメソッドをその他のターミナルに持って行って処理することもできる。これは複数のワークステーションをマルチ・コンピュータとして働かせていることと同じである。この機能は、分散化されている各種データベースのリンクデータを一つのターミナルに持ってきてとりまとめるという働きを実現させるときに使われる。同一種類のリンク構造を持つデータベースを一つにまとめる場合は、リンクデータのみメッセージ・パッシングでよいが、異なった種類のリンク構造を持つデータベースを一つにまとめる場合は、そのリンクを処理するメソッドを持っていない場合もあるので、メソッドのメッセージ・パッシングを伴うこともある。



(図8) 拡張メッセージ・パッシング

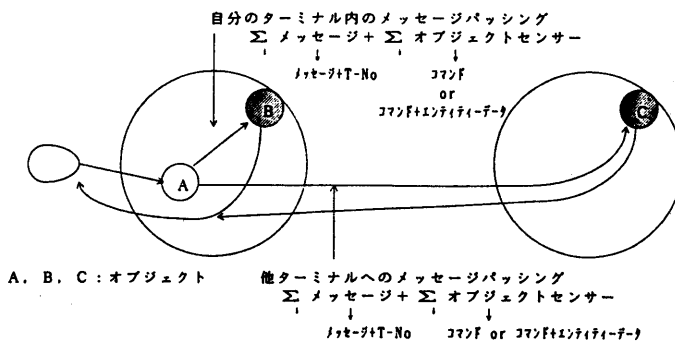
分散処理をさらに使いやすくするためにメッセージ・パッシングの概念を図8に示すように拡張する。ターミナル X_1 と X_2 にインストールされているオブジェクトにおいて同一のオブジェクト同志がやりとりをして、分散処理をスムーズに動かそうとする方法と、メッセージを相手ターミナルのオブジェクトに渡して処理を続行させる方法とである。前者を同一オブジェクト・メッセージパッシングと呼び、後者をシーケンスオブジェクト・メッセージパッシングと呼んでいる。たとえば、“ファイル代行”という機能は自分のターミナルにメモリが少ないとき、他ターミナルのメモリを代わりに使わせてもらおうというものである。これはメッセージ・パッシングでいえば、あたかも自分の中にあるデータのようにみせて他ターミナルのデータをどこかのターミナルで処理してもらおうというものである。“ターミナル代行”という機能もあたかも他ターミナルを自分のターミナルのように思わせようとするメッセージ・パッシングである。

同一オブジェクト・メッセージパッシングのやりとりは、どのターミナルとの間で行うのかが分かっているなければならない。この問い合わせを行う方法には、管理センター方式とブロードキャスト方式が考えられる。前者は、分散システムの中にデータ管理のための集中処理をするセンターを設ける方法であり、後者は共通線信号方式や無線を使ったバス型LANなどのネットワークを使う方法である。

メッセージ・パッシングの考えをさらにオブジェクトセンサーに適用してみる。前記したオブジェクトセンサーはオブジェクトというカプセルのインスタンスに適したものであるが、これをメソッドにも適用することができる。このことをオブジェクトセンサー・メッセージパッシングと呼ぶ。メソッド・メッセージパッシングはリモートスワッピングの概念と同じものであり、リモートスワッピング時、二次メモリに記憶してからローカルスワッピングを使えばプログラムのダウンロードが実現する。このことを“ラーニング”と呼ぶ。このことから、オペレータに対してプログラムやデータの“一時使用”、“永久使用”という概念が出現する。

3-5)分散システムにおけるメッセージ・パッシング

図9にメッセージ・パッシングのフォーマットを示した。オブジェクト間のメッセージ・パッシングは自分のターミナルのオブジェクト間のみではなく、他ターミナルにあるオブジェクトとの間にも適用される。フォーマットとしては、オブジェクトセンサーのみの転送ではなく、処理の内容(メソッドを表すコマンド)まで一緒に転送する方式である。自分のターミナル内のメッセージ・パッシングを使用して、他ターミナルのオブジェクトさえ動かすことができる。メッセージにしてもオブジェクトセンサー(コマンド+エンティティデータ)にしてもいくつも組合せることができる。



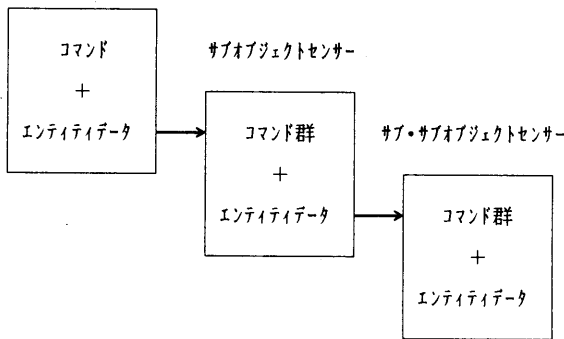
(図9) メッセージ・パッシングのフォーマット

4. ユーザ・オリエンテッド方式

システムに使用したいデータのユーザによる自由な設計を可能にするユーザオリエンテッド方式について述べる。

4-1) 情報表現モデルの階層化

実世界のデータをアクセプターにより取り込み、概念モデル、論理表現モデルとして、“オブジェクトセンサー”という仮想化が出て来たことを前述した。ここでは、さらにこの理論をすすめて、情報表現モデルの階層化を試みる。図10にオブジェクトセンサーを階層化した構造を示している。オブジェクトセンサーはコマンドとエンティティデータによって構成される。そのエンティティデータの中味としてコマンド群で取り扱ふことにより、より深いリンクをコマンドで取って行くようにしたものがサブ・オブジェクトセンサーであり、サブ・サブ・オブジェクトである。このコマンド群の間にリンクオブジェクトセンサー



(図10) オブジェクトセンサーの階層化

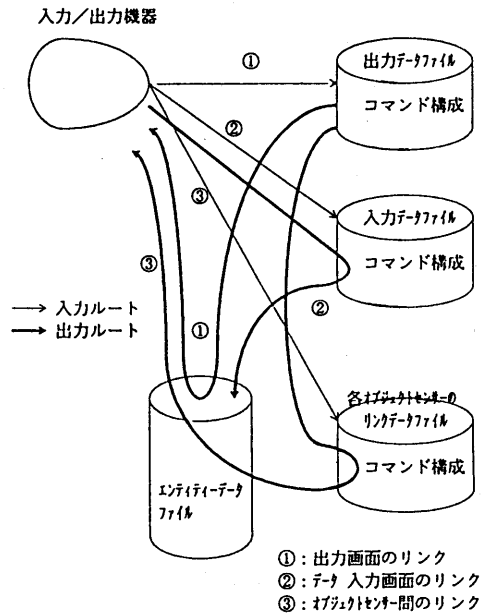
ク条件も付けられるようにした。リレーショナルリンクが基本であるが、さらに入出力フォーマットの区別や、出力の順序、ウィンドー条件、検索条件、キュー管理の条件なども付けることができる。

4-2) ユーザ・オリエンテッド方式

図11にユーザオリエンテッドを実現するためのファイル構造を示す。システムにおいて、ユ

ーザが自由に設計したいものとしては、データを入力するための入力フォーマット、入力したデータを自由に組み合わせて出力するための出力フォーマット、さらに、入力したデータの間の自由なリンク付けがあげられる。

そこで、出力フォーマットを決定する出力データファイルを設け、入力機器により出力フォーマットを入力する。データは、コマンドというオブジェクトセンサーにより入力される。



(図11) ユーザオリエンテッドなファイル構造

出力するときは、エンティティデータファイルでエンティティデータに変換し出力する。入力フォーマットを決定するには、入力データファイルを設け、入力機器により、入力フォーマットを入力する。エンティティデータを入力するときは、入力データファイルにより、入力フォーマットを出力し、そのフォーマットに従って入力する。以上の入力フォーマット、及び出力フォーマットを決めているリンクデータがサブ・オブジェクトセンサーである。

さらに、各オブジェクトセンサーに自由なり

リンクを付けるためにリンクデータファイルを設計、入力機器により、各オブジェクトセンサー間にリンクを付けるデータを入力する。出力データファイルからデータが出力されるときにこのリンクデータファイルに登録されているコマンドがあれば、そのコマンド間でリンクをとり、エンティティデータ化されて出力される。このオブジェクト間のリンク条件を決めているリンクデータがサブ・サブオブジェクトセンサーである。このセンサーのことを“リンクセンサー”と呼んでいる。

4-3) 画面による動作組み合わせ処理

“ハイパー言語”

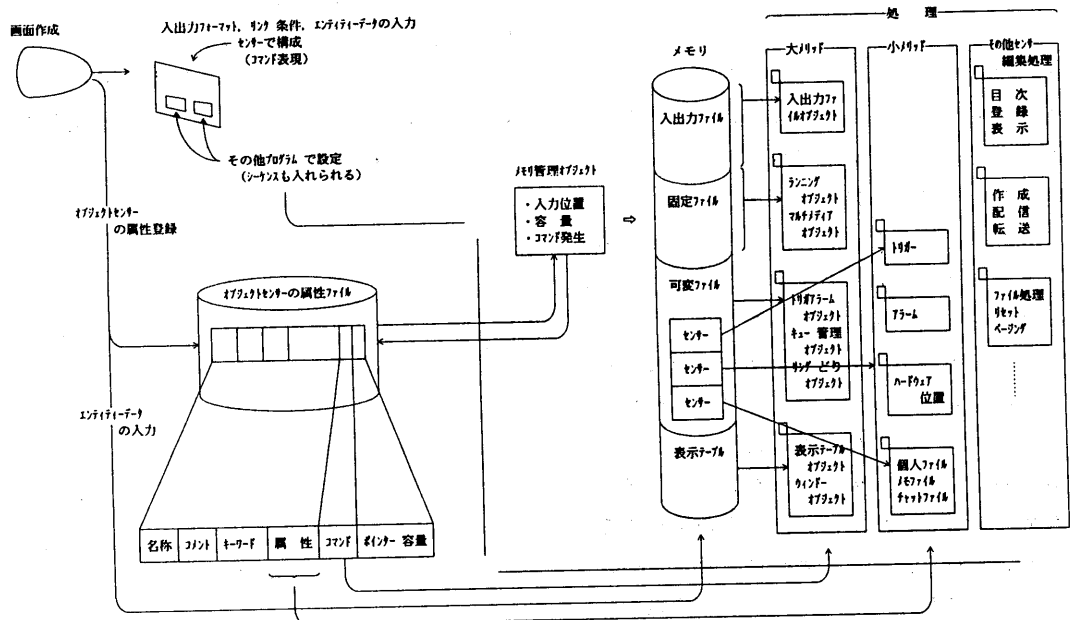
以上のユーザ・オリエンテッド方式を実現するため、本方式では図12に示すような“ハイパー言語”を用意した。これは、オブジェクトセンサーの属性データを登録し、エンティティデータを格納することによって、コマンドが各ファイルの大きなうごき（大メソッド）を選択し、属性の種類によって小さなうごき（小メソッド）が選ばれ、各オブジェクトセンサーの基本的処理が与えられる。このように与えられたオブジ

ェクトセンサーを使って、出力フォーマットやエンティティデータの入力フォーマット、各オブジェクトセンサーのリンク条件が画面から入力される。このとき、さらにオブジェクトセンサーの情報行動を決めるメソッドオブジェクトセンサーを画面に属性として埋め込むことによって、システムのシーケンスを決めることができる。このように、“画面作成による動作組み合わせ”を行う一連の操作手順のことを“ハイパー言語”と呼んでいる。

これらのうごきは、外部スキーマの高度な編集である。

5. 各オブジェクトセンサー間のリンクデータファイル“リンクセンサー”

シーケンスを組むときには、いろいろな形のデータを取り扱う場合がある。ただ単に、カード型リンクでよい場合もあるだろうし、ネットワーク型リンクの場合がよい場合もある。また、リレーション型リンクの場合が便利な場合もある。目的（オブジェクト）に応じて、その都度最適なリンクがとれるように考えたものがリンクセンサーで



〈図 1 2〉 画面による動作組み合わせ“ハイパー言語”

ある。ネットワーク型リンクやツリー型リンクは、1つ1つのリンクを作り、それをリンクテーブルに持った。このため、どんなリンクでも作れるが、追加データや変更データがあるとリンクデータを再度、張り直さなければならないため、データの追加・変更・削除がやりにくいという欠点があった。

そこで考えられたのが、ファイル構造によってリンクをとるというリレーショナル型リンクであった。この方法でリンクをとるとデータの変更・追加を行ってもリンクデータの作りなおしが不要、リンクテーブルの参照という手間もなく、プログラムを動かすだけでリンクをとることができることであった。

しかし、現在の要求は、ツリー型リンクやネットワーク型リンクの方が便利な場合や、データデータベースというより、ファイルの固まりのような時系列型リンク、さらには、ハイパーリンクのようにリンクの規則性のないものまで統合的に扱うことが必要となって来た。

今まで出て来た各種リンク（ネットワーク、ツリー、リレーショナルさらには、フレーム、ハイパー、時系列・・・）を自由に組むことができ、追加・変更・削除が簡単にできて少量、多量のデータを扱え、規則性のデータも不規則性のデータもスピードが早く、そして、さらにマルチメディアデータが取り扱え、また、それがリモートすなわち、分散で取り扱えるような統合的なリンク構造を持ったデータベースが欲しいのである。そこで、各種リンクを取るためにリンクデータの必要なネットワーク型リンク、ツリー型リンク、ハイパー型リンクのオブジェクトセンサーとしてリンクセンサーを用意した。もう1つのリレーショナル型リンクの処理は、リンクテーブル方式ではなく、構造リンク方式であるため、これを前のリンクセンサーに統一するために構造を示すフォーマットのデータをリンクセンサーに持たせるようにした。このようにすれば、リレーショナルリンクの構造ももっと便利になる。すなわち、過去、構築したあ

るリレーショナルの構造を変更することも簡単に出来る。今あるデータベースのリンクセンサーの他に新しいリンクセンサーを用意すれば、過去のデータベースはそのままにして変更した新しいリンクにデータベースを作っていくことができ、必要なときはいつでも過去のデータベースとつなぎ合わせることができる。

では、追加・変更・削除機能や大量のデータもという要望はどうするか、これこそリレーショナル型リンクの基本的考えを導入すればよい。リレーショナルデータベースを考え出した基本的コンセプトは“リンクテーブルを持たず構造によってリンクをとろう”という考えであった。そこで、根本的な考えである“構造によるリンク方式”を取った。これが、オブジェクトセンサーにおける階層化によるコマンドリンク方式、及びコマンドリンクオブジェクトの存在である。そこで、前に述べたリンクセンサーと結びつけ、あらゆるリンクはコマンドで行うようにした。従って今までのネットワーク型リンクやツリー型リンクさらに、少量のデータのためのハイパーリンクでさえリレーショナル型リンクと同じコマンドでリンク付けをやることになるためこれらも独立性の高いリンクを取り扱うことになる。

検索を便利にするということからオブジェクトセンサーの属性データすべてを検索のキーワードとする方法により、フレーム型リンクとよく似た検索が実現できる。

6. むすび

分散システムに要求される条件とその実現方式について議論してきた。特にマルチメディアデータをリモートで取り扱いを可能にする情報表現モデルとして、オブジェクトセンサー方式を提案した。このオブジェクトセンサーの属性コンセプトからコマンド発生、コマンドリンクテーブルへの展開を説明し、特にこのコマンドリンク方式からは、ファイルのフリーフォーマット化を実現でき、データのトレンドをとるキュー管理をも実現できることを示した。さらに、リンクセンサーにより、各種リンクをも統一することを示した。拡張メッセージパッシングにおけるリモートスワッピングとオブジェクト・メッセージパッシングの考えは、高度な分散処理に有効であることも示した。そして、オブジェクトセンサーの階層化、及び“ハイパー言語”の提案により、ユーザオリエンテッドな形で入出力フォーマットを自由に設定でき、オブジェクトセンサー間のリンク取りが自由に行なえることを示した。

今回は、分散システムにおけるユーザカスタマイズ可能なファイル構造についてその概念を述べた。今後は、この理論を実機のワークステーション上で実現していき、機能面、性能面に渡って検討していく予定である。なお、今後の課題として、

- ・ 目的別データベースの互換性、協調性について
- ・ 分散処理のメッセージ・パッシングのやり方、オーバーヘッドについて
- ・ オブジェクトセンサーの最適な部品化
- ・ ハイパー言語のあり方 など

について検討、実験をすすめ、順次報告していく予定である。

参考文献

1. 野口他 “マルチメディア情報処理” オーム社 (1987)
2. 村山, 龍, 高原他: “ネットワークとサービスのアーキテクチャモデルの提案” 電子情報通信学会70周年総合全国大会 1804, 1987
3. 高原: “通信サービスへの構造的アプローチ” 電子情報通信学会 通信サービス時限研究会 第2回通信サービスシンポジウム 1988
4. 龍: “通信サービスの将来像” 電子情報学会” 通信サービス時限研究会 第2回通信サービスシンポジウム 1988
5. 知的コミュニケーションと符号化ワークショップ資料 電子情報通信学会 1989
6. 高原, 龍: “理想技術論と情報ネットワークシステム” 応用科学学会誌 Vol.4 No.1 1990
7. 田中, 久保田: オブジェクト指向交換プログラムの分散環境での実行制御 情報処理学会マルチメディア通信と分散処理研究会 45-7 (1990)
8. 増永: マルチメディアデータベース総論, 情報処理28巻6号 (1987.6)
9. 増永: 分散型データベースシステム, 情報処理28巻4号 (1987.4)