

施設管理システムの開発とその運用

株式会社 オオバ 浜 出 美智雄

○吉 田 倉

高 田 和 典

■はじめに

施設管理システム(FM)はコンピュータマッピングシステムの応用システムである。管理対象となる施設には、電線・ガス管・水道・下水道などがある。広範囲に展開されているが故に、地図上にプロットすること抜きにその把握が難しい施設である。施設管理と言っても自動制御のような管理ではなく、布設位置や設計データの検索・集計などを行うことによって、事故や老朽化対策などの維持・管理業務を支援するものである。データベースマネジメントシステム(DBMS)の一種と言えるが、コンピュータマッピング技術と結びついている点が特徴である。文字、数値情報と同様に図形(地図)情報も扱うDBMSと表現できるのではないか。

コンピュータマッピングはカナダ、アメリカが先進国であり、大規模な施設管理システムが開発・運用されている。最近ではビデオなどを組込んだマルチメディア化などが研究されているようである。国内における大規模なものには、電力・ガス会社の社内システム、政令都市での道路・上下水管理システムなどがある。

本稿はある政令都市(人口150万)の下水道管理システムの開発、そして運用事例を報告するものである。

■ワークステーション(EWS)ベース

開発に着手したのは平成元年からである。事前調査、基本設計、プログラム開発と進んだ訳であるが、予想されたデータ量が約3GB(ギガバイト)もあり、会話型処理を基本とするシステムであることから、ハードウェアの選択には慎重さが求められた。従来のマッピングシステムはミニコンとGWS(グラフィックワークステーション)との組み合わせが主流であり、大型機も関与していた。しかし、開発システムはワークステーションベースとした。当時はRISC型CPUの登場により、それを採用したワークステーションの処理速度が飛躍的に向上しはじめた時であり、接続できる記憶装置の容量も増え、価格も月ごとに改定される、という状況があったからである。

ワークステーションの採用は、このようなハードウェア面の優位性からだけではなく、ソフトウェア環境における優位性も大きな理由であった。UNIXマシンとしてのOSの標準化、XWindowシステムとしてのグラフィックスの標準化、さらにはオブジェクト指向をとりいれたGUI(グラフィクスユーザーインターフェイス)の統一化とそのAPI(アプリケーションプログラマーズインターフェイス)の提供など、開発思想の国際的統一を土台にした開発環境の充実、ミニコンや大型機にない特長である。

■CMAPT-X

CMAPT-Xは汎用施設管理システムであり、施設管理システムを開発するためのツールとして開発された。グラフィックはXWindowを採用(Xlib)、GUIはOPENLOOK準拠のXViewにてプログラムされている。

下水道管理システムCMAPT-XSは、その応用ソフトである。CMAPT-Xのソースコードを書き換えたものではなく、CMAPT-Xが提供するカスタマイズ言語により記述されたものである。

■128ファイル同時表示

下水道施設情報は、下水道台帳図1枚の情報を図形と属性に分け、それぞれのファイルに納めている。図形はベクトルデータである。図形ファイルは同時に128ファイル表示することができる。表示処理に特徴があり、表示データをあらかじめメモリーにロードする処理などは行わず、ハードディスクに直接アクセスし

ている。また、複数のファイルを表示している時でも実際にオープンしているのは1ファイルであり、オープン、クローズを繰り返しながら128ファイルの同時表示を実現している。したがって、128という数字に特別な意味はなく、増やすことも可能である。

このような、一見複雑な処理をしながらも、表示は高速である。クリッピング処理などをグラフィックスシステムにまかせず、アプリケーション側で対応していることの効果が表れている。また、メモリーロードがないため、瞬時に表示が開始されることも、操作性の向上につながっている。

128ファイル同時表示機能は、上下流追跡機能などにとって必須の機能であり、広範囲を表示する必要のあるコンピュータマッピングシステムの特徴的機能と言える。

■スームスクロール

図形データがベクトルデータでありながらも、高速で滑らかなスクロール機能を実現している。スクロールは8方向が可能であり、どのような操作の途中であっても行うことができる。128ファイル同時表示していても、拡大している下でのスクロール速度はほとんど低下しない。

スクロール処理はそう複雑ではない。XViewのスクロールバー機能が、指定した幅だけ描画データをずらしてくれるので、新しくウィンドウに現れた矩形エリアを再描画する処理を繰り返すだけである。XViewが教えてくれる再描画エリアを高速に描画できさえすればスームなスクロールが実現される。高速な描画は、描画範囲にある図形レコードをいかにすばやく発見するか、にかかっており、これは検索アルゴリズムの問題である。特別なグラフィック処理が関与しているわけではない。

■フォント表示・シンボルライン表示

下水道台帳図は文字データで埋められている。また、人孔をはじめ、図式に従ったシンボルが多種類存在する。さらに、法敷面を表す特殊な線種のラインストリングなども現れる。これらはマッピングデータの特徴であり、ベクターフォント登録機能と表示処理がマッピングシステムの汎用性と性能に大きく影響する。

CMAPT-Iはベクターフォント登録機能を持ち、ラインやラインストリングで作図することにより、128点を上限とするフォントデータを登録することが出来る。フォント表示は、ダイナミックなメモリーローディングが行われ、あらかじめ全データをローディングすることによるメモリーの無駄使いを避けている。

シンボルラインは2個のフォントの連続として表現される。例えば、鉄道線路は黒と白の矩形シンボルが交互に並んだものとなる。登録は、線種コードに二つのフォントコードと、それぞれの幅・高さを定義しておくだけである。

フォントデータはコードとしてデータファイルに格納される。また、シンボルラインはフォントコードも格納していない。シンボルラインと通常のラインストリングは同じデータ量である。シンボルラインを構成する線要素をファイルに展開するような愚はおかしていない。

■図形処理言語

いわゆるマクロ言語であるが、入出力命令(GET, PUT), 制御命令(IF, DO, WHILE), メニュー関数などの様々な特殊関数が備わっており、データ入力、修正、そして管理システムの検索、集計、出図処理など、ほとんどの業務を記述することができる。下水道管理システムは約200本のプログラムから構成されておりCMAPT-Iがインタープリターとなって数々の機能を実行してくれている。

この言語がサポートされた事により開発の分散化が促進され、開発者が互いに悪影響を及ぼしあうような開発上のトラブルが回避された。CMAPT-I本体とリンクする必要はないし、それぞれのプログラムが完全に独立しているからである。また、この言語で実現できない事が発生した場合は、CMAPT-I本体を拡張し、あらたな命令や特殊関数を言語仕様に加えることになるが、このような開発手法により、オーバースペックに

よるソフトの肥大化や複雑化に陥ることのない拡張が実現できている。細かな処理を本体に組込む必要がないからである。

■属性構造定義

コンピュータマッピングは一般のCADシステムのように図形処理に目が向けられがちであるが、高度な属性データ処理と結びついてこそ高付加価値を得ることができる。施設管理システムの場合はなおさらである。CMAPT-Iは属性データ処理を深く組込んでいる。オーバースペックを排したコンパクトなDBMSをめざしたが故であり、図形データとより直接的な関連を持たせたかったからである。

CMAPT-Iの属性データベース構造は、RDFと呼んでいるテキストファイルにその構造を記述することで自由に定義できる。

図形データと属性データは別々のファイルに格納されているが、そこではレコード間の関連が大切となる。CMAPT-Iでは、双方が互いのファイル内レコードアドレスを持ち合うという直接的な方法を採用している。この方法は、片方の検索だけで主題図が作成できるなどの利点をもたらす。片方のレコードが移動したりすると、即座に関係は断ち切れてしまうことになってしまうが、図形・属性の両データを扱うDBMSであるCMAPT-Iは、これを保持しながらデータ登録、削除、修正、検索、集計を行っている。

■属性処理言語

属性データへのアクセスは、問い合わせ言語SQLに似た独自の言語により行っている。例えばRDFファイルを読み、構造を知る命令がSCHEMAであり、ファイルをオープンするのがINPUT命令である。検索条件設定はWHERE、出力項目の選択はSELECT命令で行う。

この言語は前述の図形処理言語からも、C言語によるプログラムからも使用することができる。また、バッチ処理やデータメンテナンスのため、この言語を解釈し実行するインタープリター (CMT-RDBS) も用意されており、調書の出力などに活用されている。

■コード定義言語

施設属性はそのほとんどがコード化されて入力される。属性の出力に際しては、これを元の値 (文字列) に変換する必要があるが、その登録を容易にし、変換を高速にするための「コード登録言語」がCDLと名付けられサポートされている。CDLは、コードの構造 (大中小分類) と変換の方法 (CONVERSION, DIRECT, SCAN) を宣言し、その後にコードとそれが意味する文字列を記述する構造になっている。

このような属性構造もコードもすべて外部ファイルにて定義できる機能は、汎用性を高めるとともにデータ項目の追加に柔軟な対応を取ることができるので、システム全体の拡張性を高めることにも貢献している。

■オブジェクトデータベース

データの入力や修正などの変更処理が発生した時に、レコード内の各項目間での整合性チェックや自動設定を、特に指定せずとも暗黙のうちにこなわせる機能がある。この機能は、RDFファイルに、そのロジックを記述したファイル名を記述しておくことにより働く。データベースには、様々な目的と方法で書き換え要求がだされるが、基本的なデータ間の整合性チェックを最下層の段階でおこなわせようとするものである。これは、上流側処理の手続きからデータチェック処理部分を省いて、簡潔にすることができるし、各手続きから独立したチェック処理は、データの持つ性格自身からそれらの手続きの誤りを指摘することにもなる。データベースのデータが、施設という実体を表すものとすれば、手続によって自由に解釈、変更されてよいはずがないのであり、そのチェック処理は、構造定義とともにRDFにて宣言されるべきであろう。「オブジェクトデータベース」指向をこのように捉え、様々な試行を行っている。

■運用事例

政令都市における下水道管理システムの運用事例を報告する。

- 人孔、管渠の布設替えの資料として、施工年次色分け図や工事年度別集計調書に利用
- 窓口業務における問い合わせ箇所の現地状況を確認するため、町丁目検索や目標物検索機能により台帳図を検索し、画面での確認及び出図に利用
- 設計業務を委託した業者や、下水道に関する解析、シミュレーションソフトを研究している大学の研究室に、交換フォーマットにて現況デジタルデータを提供
- 設計委託、施設点検委託、台帳更新委託の設計書と仕様書作成の基礎資料として、工事年度別、幹線流域別、管渠端末別等の各種集計調書、及び1/2500程度の縮尺による広域図作成に利用
- 施設改修計画の立案資料として、行政区別、処理・排水区別の各種集計調書及び1/10000程度の縮尺による広域図作成に利用
- 溢水対策の資料として、上下流検索、面積集計及び1/2500程度の縮尺による影響範囲図出力に利用

■おわりに

以上述べてきたことは、私共の開発した施設管理システムの紹介であるが、それにとどまらず、施設管理システムと呼ばれるものが保有すべき機能を提案したつもりでもある。コンピュータマッピングは、システムを開発するメーカーにとって、機械設計や電気設計部門とくらべるとその市場が狭い。そのため機械設計CADを転用するような形で供給されてきた面も否めず、相対的に遅れた分野になってしまっている。私共が、事前調査にとどまらず、システム開発にまで手を伸ばした（伸ばさざるを得なかった）原因はここにある。

システム開発を本業とはしない私共が、それでも政令都市で実稼働できるシステムを開発しえたのは、総合建設コンサルタントとして、下水道管理システムの対象についての知識が比較的にあったこと、及び小回りのきく、小さな開発体制でのぞんだことが挙げられるように思う。

開発は事前調査報告書の検討から含めて、事実上3名でおこなった。データ入力費用が6億円を超えるデータベースシステムの開発スタッフが、僅か3名とはいかにも少ないように思えるが、人手不足であったからではない。数十人のチームを組まなければ開発できないものもあろうが、人数が多ければそれだけ仕様打ち合わせと調整のために無駄な時間が費やされる。開発期限が許してくれるなら、開発はできるだけ小人数にするべきである。それを実証できたと考えている。トップダウンとボトムアップをいかに円滑に回転させるか、この点がプログラミング工程管理の核心と思うが、スタッフの規模を最小限にすることが忘れられてはならない。

また、ワークステーションベンダーから提供されている開発環境が、工数の大幅な縮小へ貢献していることも強調されなければならない。プログラミングの分野におけるオブジェクト指向の効力があらわれていると思うからである。

開発の成否は仕様にかかっているが、オーバースペック、「完璧な仕様書」には注意しなければならない。ユーザーと開発側との過度に細かな仕様協議はお互いに不利益をもたらす。仕様書の神格化というべき誤りを犯してはならない。存在しているシステムの解説書を書くことすら難しいのに、存在しないシステムの完璧な仕様書は不可能である。効率良く開発するには、試作品をつくり検討しあいながら進めるプロトタイプング技法が効力を発揮する。開発期間中、ユーザーはモニターとして開発に加わるのであり、極めて具体的に問題点があらわれ、また、余分な機能は影を潜めていくことになっていく。ソフトが専門家の特殊技術であった頃は難しかったであろうが、ユーザーが開発も担う時代になっている今だからこそ活用したい方法である。