

執筆者Profile



1980年 日本鋼管ライトスチール(株)入社。
1989年 原価計算システム開発を担当。
1990年 建材製品生産管理システム開発を担当。
1993年 物流管理システム開発を担当。
1997年 情報ネットワーク構築に従事。
現在、システム推進室課長。

松崎 豊

論文要旨

グループウェアが導入されパソコンがユーザに行き渡ったが利用レベルはまだ低い。ユーザのリテラシ向上を図る手段の一つにEUCを位置付けた。

「スキルの低いユーザにも使わせたい」「開発・投資コストを抑えたい」「運用負荷を高めたくない」などの課題を踏まえ最適なEUCを模索した。

LISTVIEWによるEUCを実施するが新しい切り口のデータを求めるユーザに支持されない。そこで以前から突発的なデータ要求に応える手段としていたクイックレポートをベースに構築を行う。ホストデータを切り出し直接パソコンに送りExcelのピボットテーブルで分析/加工をさせた。

サーバを介さないこの方法はとても単純な構成でホスト側も最小限の機能しか持たない。しかし、このシンプルさが「初心者にも分かりやすい」などすべての課題を高レベルでクリアすることにつながった。

本論文はメインフレームによるEUCシステムとシンプルゆえの強みを述べる。

論文目次

1.はじめに	《 1 》
2.開発の経緯	《 1 》
3.基本構想	《 2 》
3.1公開データの選定	
3.2データの公開方法と課題	
3.3パソコン側のデータ加工ソフト	
3.4データ転送	
3.5ホスト処理	
3.6技術的な背景	
4.システムの概要	《 6 》
4.1ハードとソフトウェアの構成	
4.2データの格納	
4.3転送データの条件設定	
4.4バッチジョブのキック	
4.5バッチジョブ	
4.6ファイル転送	
5.シンプルなシステムを目指して	《11》
6.評価	《11》
7.今後の課題	《12》
8.むすび	《12》

図表一覧

図 1 クイックレポート例	《 3 》
図 2 ピボットテーブルレポート使用例	《 3 》
図 3 システム概略図	《 7 》
図 4 レコード構成	《 7 》
図 5 条件設定画面	《 8 》
図 6 ファイル転送画面	《10》

1. はじめに

当社は、昭和35年創業の鉄鋼製品の二次加工メーカーである。製品は約50品種に及び、建築・土木・道路など多方面にわたりご利用いただいている。身近なものでは、ビルの床材に使用されるデッキプレート、道路に設置されるガードレール・ガードパイプ・防音壁などがある。

当社は2年ほど前にグループウェア/ Lotus Notes (以下Notesという)を導入し、情報共有化を目的にクライアントパソコンの台数を増やしてきた。平成11年初めまでに約450台のパソコン導入が終わり、ほぼホワイトカラーの一人1台体制ができ上がった。しかし、その利用レベルは決して高くなく、投資に見合った効果が出ているとはいえない状況である。ユーザのリテラシ向上を図りパソコン利用を活性化させるために、ワークフローの展開、LISTVIEWによるペーパーレス化、WWW利用環境の整備とともにEUCを位置づけた。

当社の基幹システムはメインフレーム上で稼働している。平成3年から受注・生産管理・物流の中核的システムを根本から見直し再構築を行った。またこれに合わせ新製品の追加や改廃で崩れかけていた品名体系の見直しも行った。これらの条件が揃いホストの中には体系的に整理可能な様々なデータが蓄積されてきた。

このデータを活用しEUCを推進するため当社に最適なシステムを模索した結果、我々はメインフレーム中心のEUCを選択した。サーバによるデータ公開が当たり前の時世に一見時代錯誤に思える。しかし、ユーザニーズと公開データの性質を十分把握した上で開発に臨めば、サーバシステムのような高機能は無理としても、使い易さにおいては決して劣らないシステムが作れると判断した。

本論文はメインフレームを使ったEUCシステムを中心に、シンプルなシステムの強みについて述べる。

2. 開発の経緯

(1) LISTVIEWを使用したEUCシステム

当初、ホストのデータをパソコンに渡す手段として、LISTVIEWを使用することを計画した。LISTVIEWには帳票からデータを切り出すための豊富な手段が用意されており、これを利用すれば現在の帳票がそのままデータソースとして利用できる。システム部門では新規に開発する部分が少なく即効性が期待できる。

しかし、結論からいえばこの方法はユーザの満足を得られなかった。ユーザの要求は既存の帳票データをパソコン上で加工することよりも、既存の帳票にはない新しい切り口のデータを手に入れることにあった。スキルのあるユーザなら切り出した帳票データを組み合わせ目的のデータを作り出せるかもしれない。しかし、パソコンを始めたばかりのユーザにそれを求めるのは無理である。

ペーパーレスとパソコンへの2次入力軽減にLISTVIEWは間違いなく威力を発揮する。だが、当社のユーザが求めるEUCを実現するには違う手法を使ったシステム構築が必要となった。

(2) クイックレポート型のEUCシステム

これまでシステム部門では、ホストに蓄積されたデータを「クイックレポート」という形で帳票に打ち出しユーザに提供してきた。帳票サンプルを図1に示す。これはある程度打ち出しのパターンを決め、その範囲内でユーザの要件を満たす帳票を作ろうとするもの

である。システム部門ではキーとなる項目の編集条件だけを修正すれば比較的簡単に帳票が作成できる。早ければ30分、遅くても翌日中には結果をユーザに提出してきた。この方法は欲しい情報が早く、簡単に手に入るとユーザにはとても評判が良かった。しかし、いくら簡単な修正とはいえ、不定期に上がってくる要望を短時間にこなすのは大変な重荷であった。

そこで、クイックレポートと同等の処理結果をユーザ自身で作りに出せるEUCシステムが構築できれば、ユーザの満足とシステム部門の負荷軽減が両立できると考えた。

問題はその実現方法である。ユーザがパソコンを使って自由にデータを加工したいといってもイメージのみ先走りの感が強い。データを渡せば本当にそこから自分の欲しいものが作れるのか、更には、自分の欲しいデータをホストから取り出すための条件を考えられるのか、など突き詰めていくとかなり怪しいことに気付く。

パソコンに取り込むデータの条件をどのように設定させるか。

取り込んだデータをパソコン上でどのように加工させるか。

この2点に注力し初心者にも「簡単!」と感じさせる仕組みを作る必要がある。ここで難しいと思われては、万人向けのはずのEUCがほんの一部のスキルのある人間のみのもので終わってしまう。

ホストオンライン画面に向かってデータを入力したり、表計算ソフトで資料を作ったりしている、いわゆる普通にパソコンを使える人をターゲットに、本当に戦力として使ってもらえるEUCシステムを目指して開発に取りかかった。

3. 基本構想

3.1 公開データの選定

ホストに蓄積されたデータのうち、第一ステップとして売上・出荷データを公開することにした。このデータはクイックレポートの対象として最も多く要望されている。営業部門では特約店やプロジェクトごとの売上数量や金額の把握、製造部門では製品や材料需給の基礎データ、あるいは、製造設備改造の検討データと広範囲に利用されてきた。年間約50万件のデータ量がある。これまでのクイックレポートは過去3年間と処理日の前月末までのデータを対象にしていた。今回のデータ公開にあたり一層の利便性を図るため、処理日の前日出荷分までを公開対象とすることにした。

3.2 データの公開方法と課題

様々な事例をみると、EUCのデータ公開には専用のサーバを使用するのが一般的なようである。サーバからODBC経由のクエリーやSQL、専用ツールを利用して項目や抽出条件を設定し、パソコンにデータを取り込んでいるシステムが多い。

しかし、当社の現状を考えるとこのような方法が必ずしも最良とは思えない。主な理由は次のとおりである。

パソコン導入から日の浅い当社ではユーザ部門にEUCの核となる人材がいない。ユーザの身近に気軽に相談できる人がいない状況でクエリー/SQLなどの教育を行っても、



図 1 . クイックレポート例

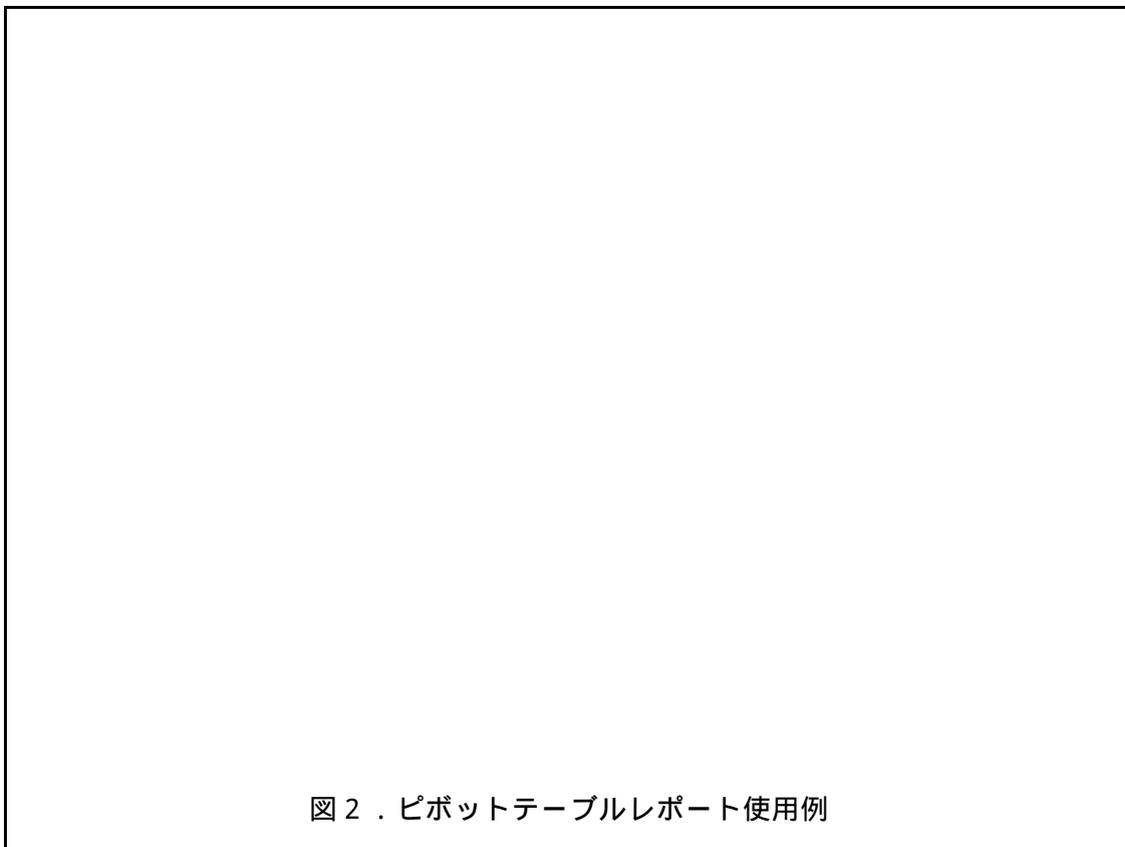


図 2 . ピボットテーブルレポート使用例

おそらく定着しないだろう。

これまでシステム部門ではメインフレームを使ったシステムばかりを作ってきた。

サーバシステムの経験は少なくスキルがない。本当に使いものになるシステムが作れるのか確信がない。

システム部門ではパソコンへのクライアントソフトインストールやサーバ管理など、新たな仕事を背負い込むことになる。

厳しい経営環境の中、新規のハード・ソフトへの投資は極力抑えたい。

このような事情から我々が考えたのは、手持ちのハードとソフトと、そしてノウハウを使ってEUCシステムを作り上げることだった。

3.3 パソコン側のデータ加工ソフト

ホストから受け取ったデータを編集・加工させるために、Microsoft Excel（以下Excelという）を使うことにした。これは全社共通の表計算ソフトとして既にパソコンにインストール済である。パソコン導入時にユーザに基本操作の教育を行っている。EUCである以上、受け取ったデータはどんなソフトを使って編集しようがユーザの自由である。しかし、それができるのは一部のパソコンに精通した人間に限られるだろう。このシステムが普通にパソコンを使える人を対象にしている以上、渡したデータの使い方まで考えておくのは必須である。

Excelの中でも我々が注目したのはピボットテーブルレポート（以下ピボットという）と呼ばれる機能である。本システムで作成した使用例を図2に示す。ピボットを使うと、Excelのシートに羅列された情報をもとに2次元の集計表がいとも簡単に作成できる。基本データの中から縦方向と横方向のキー項目と、集計したい項目（金額や数量）をドラッグアンドドロップで選ぶだけで設定は完了である。ボタンを押すと条件とおりの集計表がシート上に作成される。更に、二つ以上のキー項目を設定すると自動的に小計が計算されたり、集計結果の金額と数量から平均単価を計算することもできる。富士通のデータウェアハウス製品にもこれに似た手法を用いたものがあることから、データの分析/加工ツールとしては非常に有効な方法といえる。

Excelは約6万5千件のデータを扱える仕様になっている。目一杯使うとなるとレスポンスの悪化やメモリ不足発生の可能性がある。しかし、列方向の長さにもよるが1万件くらいであれば十分実用になる。データを絞り込み一次集計した結果をパソコンに渡すなら、これくらい容量があれば大抵のデータを納めることができる。

このようにデータベースソフトを使わず、普段親しんでいる表計算ソフトで二次加工ができる意義は大きい。データベースソフトは普通のユーザにとって、まだまだ難しい別格の存在と思われる。懸命に勉強してやっと使えるようになったいつもの表計算ソフトで「こんなこともできたのか！」となれば自然と興味を持つに違いない。

3.4 データ転送

ホストからパソコンへデータを送るにはWSMGRのファイル転送機能を使用する。WSMGRはWINDOWS/パソコンをホスト端末として使用するには必須のソフトであるため既に各パソコンに導入されている。WSMGRのマクロ機能を使って転送処理を自動的に実行することにし

た。マニュアルではこのような使い方は動作保証外と説明されているが、それほどシビアな業務ではないので異状があったら再実行して対処する。

転送に使用するファイルはCSV形式とした。この形式がExcelのみならずほかのソフトで使う場合でも最も汎用性があると思われる。転送先は誰にも馴染みのあるMy Documentsフォルダにした。ここに”出荷データ.csv”というファイル名で保存する。

3.5 ホスト処理

ホスト側の処理は次のような構成とした。

ホストオンラインの入力画面から対象とするデータの抽出条件と項目の編集条件を入力させる。

既存のオンライン・バッチインタフェースを使って入力された条件をパラメタにしてバッチジョブをキックする。

起動されたバッチジョブでパラメタに従ったデータ抽出と項目編集を行う。続いてSORT/MERGEユーティリティで分類集計を行い結果をCSVファイルに変換する。

ここで最大の問題となるのは、抽出や項目編集条件を入力するオンライン画面の構成である。ユーザはこの画面に向かって自分の欲しい情報をオーダーする訳である。わざわざクエリー/SQLを捨てたからには、普通のユーザに理解してもらえる内容でなければ意味がない。幸い、システム部門には多数のクイックレポートを作成してきた実績があり、ユーザのほしがる条件はある程度予想がつく。「すべての条件に答えられる」とはいえないが、ユーザの80%が満足できる条件は揃えられたと思っている。これより細かい絞り込みはオートフィルタなどを使いパソコン上でやってもらえばよい。少し多めにデータを送ったところで余計な通信費がかかるわけではない。

もう一つの問題は処理スピードである。上記の構成ではバッチ処理の終了をユーザに伝えるのが難しい。実際の運用では処理を起動してから5分ほど待って転送を開始するように説明する。ホスト側の処理に長時間かかると、転送ファイルができ上がる前に転送マクロが起動される恐れがある。

3.6 技術的な背景

本システムが成立するには「安価で高速な通信回線」と「パソコンの高性能化」が欠かせない。

ホスト側では最小限の集計作業しか行わないため、パソコンに渡すデータ量は多いときには2MB以上になる。これまでホスト回線の主流だった低速の専用線や従量制の PACKET 通信では、伝送時間や通信コストの問題から遠隔地のパソコンに大量のデータを送ることは難しかった。フレームリレーのような廉価な定額制で常時接続可能な比較的高速の通信回線が利用できるようになって、初めて実用可能となったのである。

データを受け取るパソコン側も高性能化が著しく、数万件のデータを表計算ソフト上でストレスなく扱えるようになった。以前ならデータ処理効率の良いデータベースソフトでなければ荷が重かった。

このように集約された処理結果ではなく、より原始データに近いデータをユーザに直接渡し加工させる技術が安価に入手できるようになった。ユーザは最小公倍数的なデータを

取り出し、フレキシブルに切り口を変え目的のデータにたどり着くことができる。

4. システムの概要

システム概略図を図3に示す。

4.1 ハードとソフトウェアの構成

ホストは富士通のM-1600モデル6、OSはXSPであり熊谷（埼玉県・本社工場）に設置されている。ホストはFNAゲートウェイサーバをとおしてTCP/IPのネットワークに接続している。熊谷にはこのほか、Notesサーバ、LISTVIEWサーバ、財務システムサーバが設置されている。遠隔地の工場（津、福山、福岡）と支店（全国25カ所）は定額制のフレームリレー網でつながっている。メッシュ型ネットワークによりホストオンラインは熊谷、Notesは東京に設置されたサーバに接続される。現在のネットワークはグループウェア導入時にDDX-PとINS-Pから切り換えた。

パソコンは全体で約450台ありOSはWINDOWS95/98である。このうち約120台にWSMGRがインストールされホスト端末としても使用されている。

ホストの開発言語はYPS/COBOLを使用している。これはシステム再構築の際に富士通から紹介されたもので非常にメンテナンス性がよい。本システムもこれを使って作成した。今回のシステム開発にあたりこれらのハード・ソフトには一切手を加えていない。

4.2 データの格納

公開用のデータはホストのDASD上に置く。クイックレポート用に用意しておいたデータはSAMファイルに格納されていた。バッチ作表であればそれでも問題なかったが本システムでは処理速度の面で不十分である。検討の結果必要最小限のキーを追加しVSAM(KSDS)に格納することにした。

キーを利用することで余計なレコードの読み込みをスキップできる。また、たとえ全データを読み込むような条件の場合でもシリンダ（コントロールエリア）単位の先読みを利用すれば、SAMの半分程度の時間で処理できることが今までの経験で分かっている。

また、格納済のデータより大きなキーを書き出す場合、拡張モードでオープンが可能である。このクラスタには毎月4万件以上のデータが追加されるが拡張オープンを使うことで保存期間の3年間、再創成が不要になる。年度ごとにクラスタを分けて作成しておけば1年に一度、3年経過したクラスタを削除し、そのスペースを本年度用のクラスタに割り当てるだけでよい。大量データファイルの再創成は煩わしい作業であるが、それが不要になる。

レコード構成を図4に示す。キーには月と支店コードを設定した。一連番号とはキーの重複を防ぐために単純に月内で1番から振られた番号である。営業部門からの要求は通常自分の支店のみに限られる。支店コードが指定されれば12回のスタート命令で1年分のデータが読み込める。営業本部や工場からは全支店対象の要求が多いため期間の指定がなければ全データを読むことになるが、そのような処理は頻繁に発生しない。

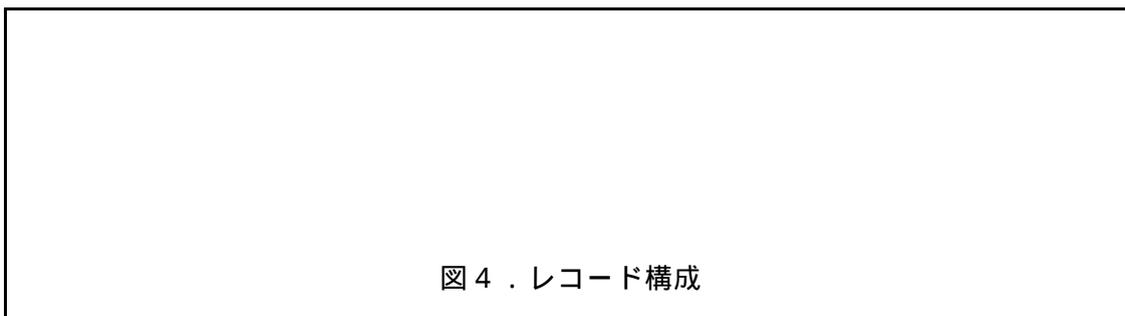
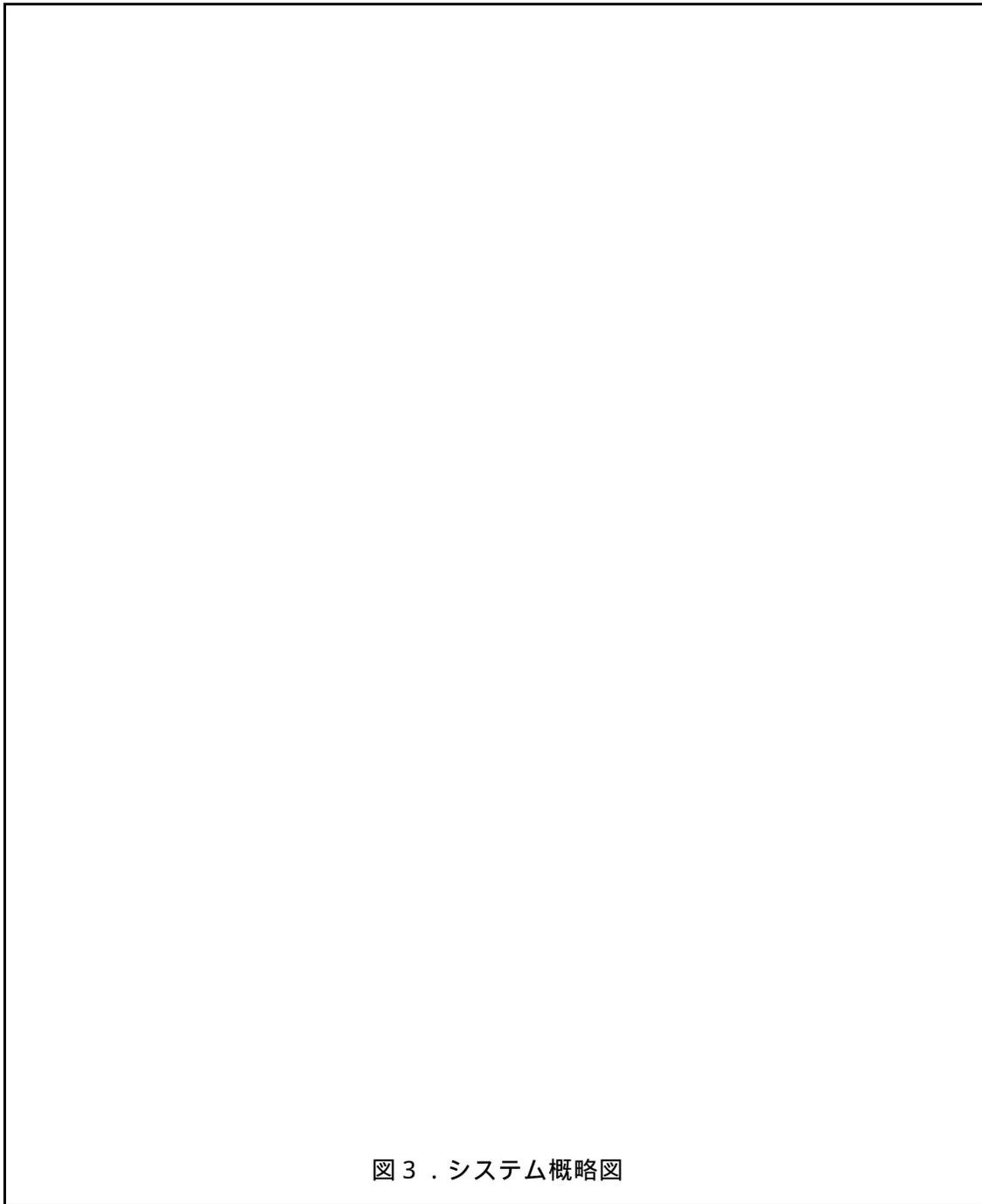




図5 . 条件設定画面

4.3 転送データの条件設定

転送データの条件を設定するオンライン画面を図5に示す。画面は大きく「処理条件入力」「抽出条件入力」「項目編集条件入力」の3部分に別れている。

(1) 処理条件入力

雛型番号とは、入力を軽減するために標準的な設定を初期表示させるために使用する。コードを指定しファンクションキーを押すとコードに対応した設定が表示される。最初はシステム部門で考えた組み合わせをいくつか用意した。今後はユーザからの要望で随時追加する。

プリンタには、自分の身近にあるプリンタの名前（APSのライタ名）を入力する。これは帳票出力のためではなく、パソコンに転送するCSVファイルのホスト上の名前として使用する。処理要求が同時に発生するのに備え、要求ごとにCSVファイルの名前を分けるためである。転送処理時にここで入力したプリンタ名を指定する。

(2) 抽出条件入力

ここでは、パソコンに転送するデータの抽出条件を設定する。クエリー/SQLを使えば=、NOT=、>、<など様々な条件が作れるのであろうが自由度の高いことが、かえって普通のユーザに敷居の高さを感じさせていないだろうか。ここでは単純明快、入力されたコードと等しいデータが抽出され項目間はすべてAND条件である。これが普通のユーザの一番自然な考え方であろう。

(3) 項目編集条件入力

ここでは、パソコンに転送するデータの項目編集についての条件を設定する。ここでも、指定の方法は単純明快である。不要な項目にはゼロ、必要な項目には「1」または編集条件を表す数字を入力する。編集条件の例を府県コードで説明すると

「1」なら府県レベル ----- 埼玉県や東京都など

「2」なら市区町村レベル ---- 埼玉県熊谷市，東京都中央区など

といったようにデータの細かさが変化する．項目の性質にあわせ様々な編集条件を用意した．凝ったことができるわけではないが通常の使用には十分であると判断している．それよりも問題は，ユーザが項目の意味を正しく理解できるかである．雛型を用意することで，項目の内容に詳しくなくてもデータが取れるように工夫した．しかし，きちんと意味が理解されないとせっかくのデータも生きない．これはユーザに勉強してもらうしかないため項目解説マニュアルを作成することにした．

4.4 バッチジョブのキック

オンラインからバッチジョブをキックするのに，当社ではオンラインプログラムでパラメタを含んだDISPLAY命令を発行し，これをSCFで拾い上げバッチジョブを起動する方法を取っている．XSPの世界では一般的な方法であると思う．今回もこの方法を取ろうとしたがパラメタが長すぎてSCFで拾いきれないトラブルが発生した．画面に多量の項目を配置し，そのほとんどをバッチジョブに引き渡すこの処理ではパラメタがどうしても長くなる．パラメタを受け渡す専用のファイルを用意することも検討したが，パラメタの多くが0と1であることに着目し，4つの項目を16進1桁表示に圧縮した．単純な変換であるが余分なファイルを使用することなく既存の仕組みの中に納めることができた．

4.5 バッチジョブ

バッチジョブは「データの抽出・編集」「分類・集計」「CSVファイル変換」の3ステップから構成されている．

(1) データの抽出・編集処理

データ格納の項で説明した内容に基づき極力，余分なレコードを読み込まないように配慮した．また，処理日の前日までのデータを取り込む仕組みを作った．前月以前のデータは月次処理時にVSAMに格納する．しかし，キー項目の構成と拡張オープンを使用する関係から日々，追加を行うことはできない．このため，同一デザインのSAMファイルを用意しデータを取り込むことにした．

テストを繰り返すうち、読み込みをスキップできる条件を与えたのに予想より処理時間が短縮されないことに気付いた．原因は先読みに期待して大量のバッファを与えたため，スタート命令によってスキップされるはずのレコードが大量に先読みされ，プログラムに通知されないところで負荷を高めていると推測した．全データを対象とする場合は確実に高速になるが，実際の運用ではスキップできる要求が圧倒的に多い．このため読み込み実験を繰り返し最適なバッファ量を模索した．シリンダの3分の1程度がバランス的に最良と思われた．

(2) 分類・集計

SORT/MERGEユーティリティでユーザの必要とする項目だけをキーにして分類・集計を行う．ホスト側で一次集計を行うことで転送データ量の削減が図れる．

(3) CSVファイル変換処理

CSVファイルに変換する際，配慮の必要な事項がある．

データにカンマを含めない。

数字のみや「1E」のようなデータは数値属性に変換される。

当社では品名名称にカンマを使用している。このため、カンマを「:」（コロン）に置き換えることにした。また、プロジェクト名や客先名にまれにカンマが使用されている場合がある。これも「:」に置き換える。

文字属性にしたい項目が数値属性になってしまうのを防ぐために、そのような恐れのある項目には先頭に「:」を付加することにした。Excelではファイルの読み込み時に列ごとに属性を指定する方法もある。しかし普通のユーザはダブルクリックだけでファイルを開きたいに違いない。多少データを汚すことになるが使い勝手を優先した。

コードと名称の扱いについても触れておきたい。例えば、客先のコードと名称は別々の項目として扱われることが多い。しかし、このようなデータをピボットで使うと項目数が増え、余分な小計がとられてしまい非常に使い難い。ピボットでデータを扱うならコードと名称は一つの項目にまとめるべきである。このため、名称のある項目はコードの次に「:」を付加し続けて名称をセットすることにした。

(例) 1234-01: マツザキショウテン ホンテン

また、ファイルの先頭には項目名をセットした見出しレコードを置く必要がある。項目が可変なため、これがないとどの列が何の項目なのか分からなくなる。そして、その項目名はそのままピボットの設定に利用できる。



図6. ファイル転送画面

4.6 ファイル転送

WSMGRのマクロで転送処理を作成した。画面サンプルと処理手順を図6に示す。本システムでパソコン側にインストールの必要なソフトは、このマクロとファイル転送用のパラメタファイルだけである。Notesの掲示板にマニュアルと一緒にこのファイルを貼り出し、ユーザ自身に自分のパソコンへインストールしてもらうことにした。遠隔地のパソコンにクライアントソフトをインストールするのはシステム部門の悩みの一つであるが、特別な配慮なしにクリアできた。

テスト中、WAN側のパソコンでこのマクロを実行すると時々正常に動作しないことがあった。レスポンスの違いが原因と思われ待ち時間の調整で回避することができた。

5. シンプルなシステムを目指して

本システムはシンプルさを目指し実用になる最小限の機能を考えて開発した。いつ使うとも知れない山のような道具を揃えるよりも、きちんと使い方のわかる最小限の道具を揃える方が、普通のユーザに安心して取り組んでもらえると考えたからである。例えば、オンライン画面が一つしかないのも単なるなり行きではない。1画面に納まるように必要な項目を慎重に吟味し、何度もレイアウトをやり直した結果なのである。

また、それは開発工数を抑えることにもつながった。メインフレームの技術者なら「これくらいのシステムなら簡単に作れる」と思われたに違いない。公開データの原形が存在していた幸運はあるが、開発に要した日数は12日ほどである。システム環境とデータを知り尽くした社内要員ならではの工数であるが、これが本システム唯一の費用である。

おそらく今回紹介したようなシステムが社外の専門家から提案されることは、ないのではないか。データの性格とユーザニーズに精通していなければ、機能の絞り込みは困難な作業である。選択を誤れば機能不足の「駄目システム」になりかねない。そのようなリスクを犯すなら機能満載のシステムを提案したほうが確実である。しかし、我々は企業内のシステム部門である。その強みを生かせば、低コストで使いやすい、シンプルなシステムが必ず作れるはずである。

6. 評価

クイックレポートの依頼者に本システムで作成したExcelのシートをメールで送り、自分のパソコンで作れることを伝えると必ず操作を教えてほしいと要望される。30分ほどの説明でユーザは使い方がのみ込める。集合教育は実施していないが利用者は確実に増えている。EUCは大勢の人に使ってもらえることこそ最大の評価ではないだろうか。

このようなシステムは定量的な効果をつかみ難い。そこで、ユーザに説明を行ったときに出た感想をいくつか上げてみる。

- ・自分のパソコンでこんなことができるとは思ってもいなかった。
- ・半日かけて作っていた資料があっという間にできてしまう。いったいあの仕事は何だったんだろう。
- ・すごい作ったね。

こっけいに聞こえるかも知れないが、本システムがユーザに大きな利益を与えられると確信した。また、心配していたバッチ処理時間も通常の要求であれば2分足らずであること

が確認できた。参考までに1999年8月の処理実績は119件、平均処理時間は87秒である。

システム部門においては「ハード・ソフトへの新規投資 ゼロ」「開発工数 最小」「運用負荷の増大 ほぼゼロ」が達成できた。更にクイックレポート作成作業が軽減された。帳票枚数の少ないレポートは随時本システムに切り換え、依頼者自身に作成してもらっている。

7. 今後の課題

本システムはデータ公開の第一歩である。これをきっかけに帳票でなくファイルでデータを欲しがることが増えてきた。更に提供情報の充実を図るため、ほかのデータも同じ手法で公開可能か検討を進めたい。

また、本システムはWSMGRのインストールされていないパソコンでは実行できない。しかし、当社ではその割合が全体の四分の一程度である。以前から不足の声が上がっていたが、本システム実施後は一層高くなってきた。費用面の問題があるが拡充を計りたい。

8. むすび

本システムはサーバでもなければWebでもない。トレンドな情報技術とは無縁である。このようなところで発表するには、ふさわしくないかもしれない。それでもあえて紹介したいと思ったのは、この程度の少機能システムでもユーザに歓迎され役立ててもらえることを伝えたかったからである。

最新の情報技術を駆使したシステムは確かに魅力的である。しかし、安定性への不安、短期間に繰り返されるバージョンアップ、それに伴うサポート打ち切りなど、誰にでも安心して取り組めるとは言い難い面もある。それならアプローチを変えて、シンプルに徹し、枯れた技術の中に最適なソリューションを見つけ出すことも一つの方向ではないだろうか。

以前「すべてのものは、付け加えるものがなくなったときに完成されるのではなく、取り去るものがなくなったときに完成されるのである」と書かれた本を読み、とても共感を覚えた。このシステムが完成されているとは決して思わないが、ここから取り去れるものは、もう、ほとんど残っていないと思っている。

最後に、本システムの普及に協力いただいた人事部の方に礼を述べてむすびとしたい。