

ツール研究 1

EVOLIO

IBM iとExcelで本格的な 業務アプリケーション開発

PRODUCT INFORMATION

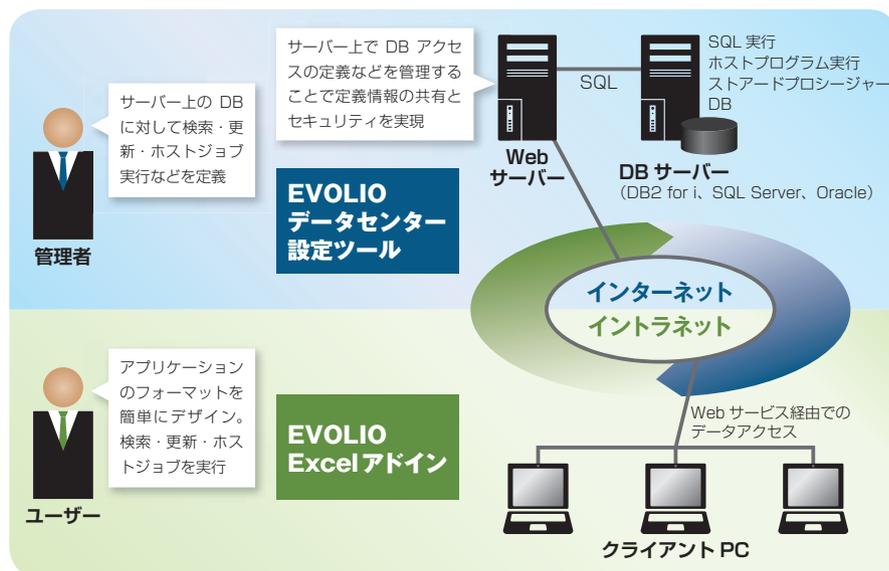
EVOLIO（エヴォリオ）は、サーバー側のDBと連携するExcelアプリケーションを簡単に作成するためのツールである。もともとは「すき家」や「なか卯」を筆頭に4147店舗を展開する外食チェーン、ゼンショーグループがグループ統合の基幹システム刷新を進める過程で開発したものだ。社内に深く根付いたExcelカルチャーをベースに開発されたExcelアプリケーションとIBM iのDB2を連携させるためのノウハウをEVOLIOに集約・標準化し、2011年3月にリリースした。



EVOLIOで作成できる アプリケーションとは

EVOLIOはExcelのアドインソフトとして構成されている（図表1）。基本的にはDBサーバー、Webサーバー、クライアントPC上のExcelにアドイン

図表1 EVOLIOのシステム構成



株式会社グローバルITサービス

<http://www.global-its.co.jp/>

TEXT

高橋 昌顕

株式会社ゼンショー
グループIT本部 TK開発室
ゼネラルマネジャー

をセットして動かす、いわゆる3階層システムとなる。

作成できるアプリケーションには、IBM iのDBとリアルタイムにデータ更新するものと、IBM iのDBからリアルタイムにデータを抽出して分析やレポート作成を行うものがある。

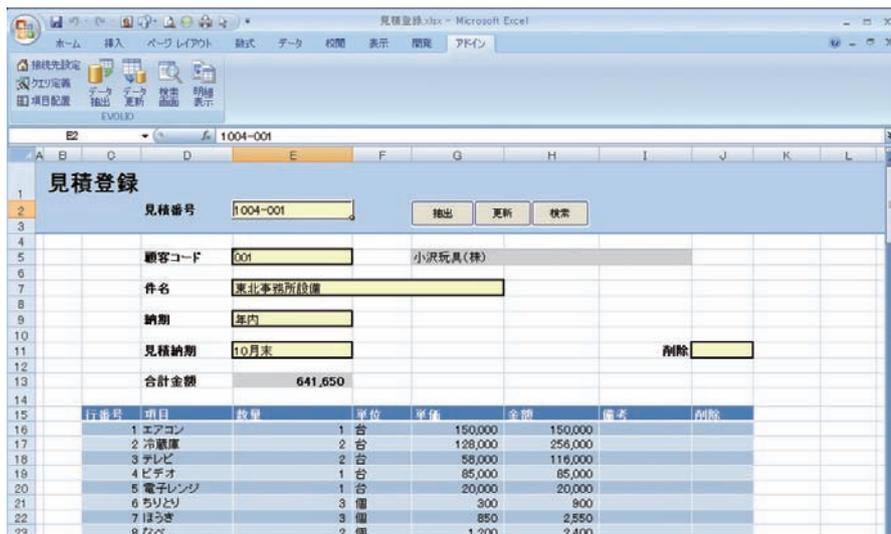
アプリケーションを作成する手順はいずれの場合も、まず「データセン

ターでDB連携を定義」し、次に「Excelのデザイン」を行う。Excelのデザインには、「入出力画面」と「レポート発行」の2種類がある。アプリケーションはExcelファイルそのものであり、直接クライアントPC上のファイルを開くことで実行される。

ここでは実際のアプリケーション作成手順を通して、EVOLIOによる開

発のスピード感を体験していただきたい。IBM iのDBとリアルタイムにデータを更新するアプリケーション例として、見積登録を実行し(画面1)、その内容を見積書として出力するアプリケーションを作成(画面2)。マスターとしては、顧客コードをキーとした顧客マスターを使用。その他の件名、商品項目、単価などは入力フリーとした。

画面1 見積登録アプリケーションの例



画面2 Excelで見積書を出力したイメージ

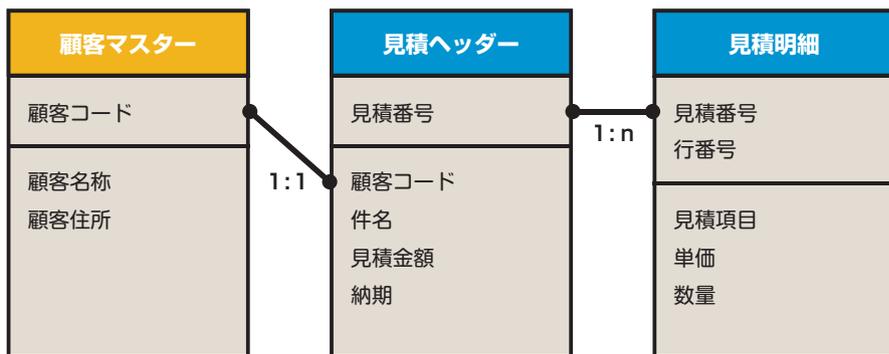


オペレーションイメージを知る 1

IBM iのDBとリアルタイムにデータ更新する例

この「見積登録」の例では、図表2のように3つのテーブルを使用する。見積内容は、見積ヘッダーテーブルに見積番号をキーとして、顧客コードや件名など1つの見積に固有な情報を登録する。見積明細テーブルには、1見積で複数の見積項目を登録するものとする。また顧客名称などの顧客情報は、顧客コードをキーにして顧客マスターで管理する。

図表2 見積登録で使用するテーブルの関連図



Step 1

データセンター設定ツールでDB連携を定義する

アプリケーションを作成するには、まずEVOLIOの「データセンター設定ツール」(以下、データセンター)でDB連携を定義する。この設定によりWebサービスが構成され、Excelクライアントに対してSOAP通信でサービスを実行する。一方、DB2側へはデータベースプロバイダーによりSQLで

連携する仕組みである。

データセンターは、クライアントPC上で動作するWindowsアプリケーションである。DB連携の定義の単位をクエリグループと呼び、1つのクエリグループがExcelの1つのシートに対応する。また、1つのクエリグループは1つ以上のクエリから構成され、データの抽出や更新をそれぞれのボタンに対応させて連続して実行できる。

画面3は、「見積登録」で使用するクエリの画面で、サーバー内のオブ

ジェクトを一覧表示する領域、対象テーブルのリレーションを定義する領域、表示・更新対象の項目の条件式や入力規則を設定する領域、クエリ全体のプロパティを定義する領域などから構成される。

また「見積登録」では、見積ヘッダーに対して抽出・更新を行うクエリと、見積明細に対して抽出・更新を行うクエリの2つを作成する。

新規にクエリを作成する場合には、画面4のように、まず使用するテーブルを左側のオブジェクト一覧からテーブルのリレーションを定義する領域にマウスでドラッグする。その後、画面5のように、複数テーブルがある場合のリレーションを定義する。この例では、見積ヘッダーの顧客コードと顧客マスターの顧客コードを対応づけて、顧客マスターから顧客名称を取得できるようにしている。

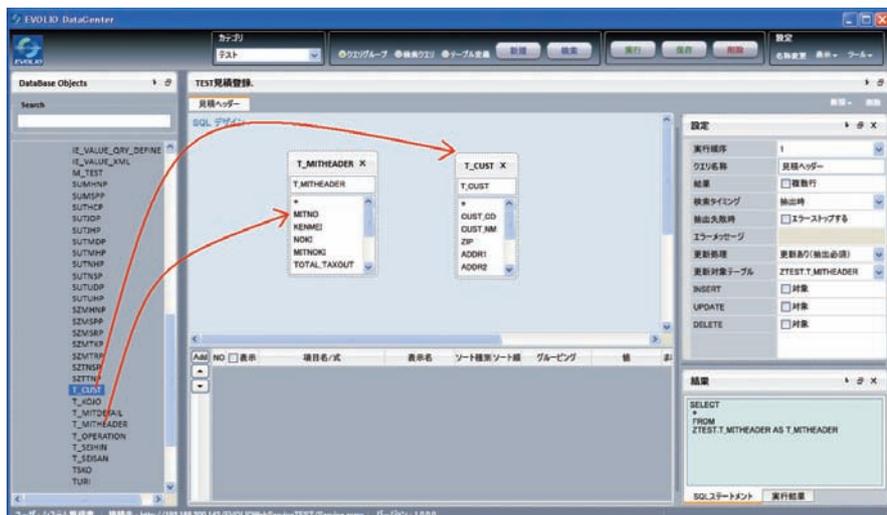
次に、テーブル内で使用する項目を定義する。

選択したテーブル内の項目をマウスでダブルクリックして選択していくと、画面6のように対象項目が設定領域に配置されるので、必要項目をすべて選択後に、抽出する条件やソート順

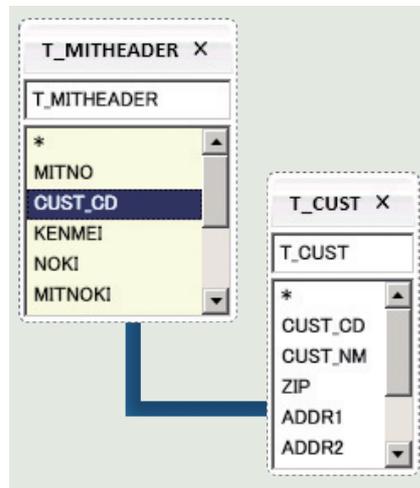
画面3 クエリの定義



画面4 テーブルの選択 (見積ヘッダー)



画面5 テーブルのリレーション定義 (見積ヘッダー)



を指定する。ここでは見積ヘッダーの主要項目と、顧客マスターの顧客名称を選択した。

項目の値として、Excel上の特定のセルと結びつけるための「変数」を定義できる。変数は、\$記号で囲まれた変数名からなり、あとでExcelのデザイン時に変数名とセルをマッピングする。「変数」は「見積番号」のように抽出条件の項目式として、あるいは「合計金額」のように更新時の値として使用できる。

あとは、クエリー全体のプロパティを設定する(画面7)。対象のクエリーが、1件ずつ抽出するか複数行を抽出するかの区分、当該クエリーの検索タイミング(抽出ボタンか更新ボタンか)、抽出に失敗した時の後続クエリーの実行可否、更新の有無と更新ありの場合の対象テーブル、更新の種類(INSERT、UPDATE、DELETE)を細かく指定できる。

以上の手順でクエリーを作成する。上記までの見積ヘッダーの定義と同様に、見積明細の定義を作成する。見積明細では、単独のテーブルに対して定義すればよい。画面8に項目定義、画面9にクエリーのプロパティ定義を載せる。これで、データセンターでのDB連携定義は完了である。

用する。

まずEVOLIOのリボンから、「クエリー定義」をクリックすると、Excelのウィンドウの右側に「クエリー定義」ペインが表示される。Excelのシート名「見積登録」を選択し、データセンターで作成した「クエリーグループ」を指定する(画面11)。

次に項目配置を行う。リボンの「項目配置」をクリックすると、XMLソースペインが現れ、クエリー作成時に表示対象とした項目および変数が表示される。ここから各項目を選択してシート上の任意のセルにドラッグ&ドロップでマッピングする(画面12)。

変数の「見積番号」をE2セルにド

画面6 使用する項目の定義(見積ヘッダー)

NO	<input type="checkbox"/> 表示	項目名/式	表示名	ソート種別	ソート順	グルーピング	値
1	<input type="checkbox"/>	T.MITHEADER.MITNO	見積番号		0	WHERE	=\$見積番号\$
2	<input checked="" type="checkbox"/>	T.MITHEADER.CUST_CD	顧客コード		0		
3	<input checked="" type="checkbox"/>	T.CUST.CUST_NM	顧客名		0		
4	<input checked="" type="checkbox"/>	T.MITHEADER.KENMEI	件名		0		
5	<input checked="" type="checkbox"/>	T.MITHEADER.NOKI	納期		0		
6	<input checked="" type="checkbox"/>	T.MITHEADER.MITNOKI	見積納期		0		
7	<input type="checkbox"/>	T.MITHEADER.TOTAL_TAXOUT	合計金額税抜		0		\$合計金額税抜\$
8	<input type="checkbox"/>	T.MITHEADER.TOTAL_TAXIN	合計金額税込		0		\$合計金額税抜\$*1.05
9	<input type="checkbox"/>	T.MITHEADER.MITDATE	見積作成日		0		CURRENT_TIMESTAMP

画面7 クエリー全体のプロパティ定義(見積ヘッダー)

設定

実行順序: 1

クエリー名称: 見積ヘッダー

結果: 複数行

検索タイミング: 抽出時

抽出失敗時: エラーストップする

エラーメッセージ: 指定した見積はありません。

更新処理: 更新あり(抽出必須)

更新対象テーブル: ZTEST.T_MITHEADER

INSERT: 対象

UPDATE: 対象

DELETE: 対象

画面8 使用する項目の定義(見積明細)

設定

実行順序: 2

クエリー名称: 見積明細

結果: 複数行

検索タイミング: 抽出時

抽出失敗時: エラーストップする

エラーメッセージ:

更新処理: 更新あり(抽出必須)

更新対象テーブル: ZTEST.T_MITDETAIL

INSERT: 対象

UPDATE: 対象

DELETE: 対象

Step 2

Excelのデザインを行う

EVOLIOのExcelアドインをインストールすると、Excelを開いた時に、アドインというタブにEVOLIOのリボンが表示される(画面10)。Excelでアプリケーションをデザインするには、「クエリー定義」と「項目配置」を使

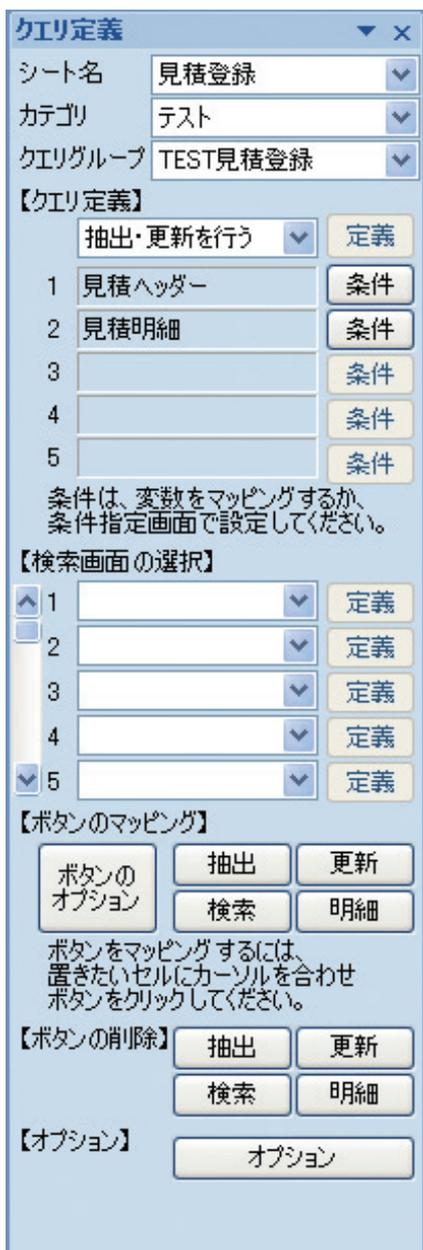
画面9 クエリー全体のプロパティ定義(見積明細)

NO	<input type="checkbox"/> 表示	項目名/式	表示名	ソート種別	ソート順	グルーピング	値
1	<input type="checkbox"/>	T.MITDETAIL.MITNO	見積番号		0	WHERE	=\$見積番号\$
2	<input checked="" type="checkbox"/>	T.MITDETAIL.LINENO	行番号		1		
3	<input checked="" type="checkbox"/>	T.MITDETAIL.ITEM	項目		0		
4	<input checked="" type="checkbox"/>	T.MITDETAIL.UNIT_PRICE	単価		0		
5	<input checked="" type="checkbox"/>	T.MITDETAIL.UMSR	単位		0		
6	<input checked="" type="checkbox"/>	T.MITDETAIL.QTY	数量		0		
7	<input type="checkbox"/>	T.MITDETAIL.AMT	金額		0		\$見積明細/単価\$ * \$見積明細/数量\$
8	<input checked="" type="checkbox"/>	T.MITDETAIL.BIKO1	備考1		0		

画面 10 EVOLIO のリボン (Excel2007 の例)



画面 11 クエリー定義



ラッグする。変数とセルがバインドされて、このセルに入力した見積番号を抽出条件にして、見積データを取得することになる。必要に応じて、シート上の任意のセルに式を入力し、合計値の表示を追加したりもできる。これは Excel を使用するメリットでもある。

次に、「抽出ボタン」と「更新ボタン」の作成だが、ボタンを置きたいセルにカーソル(複数でも可)を合わせ、クエリー定義の「ボタンのマッピング」のそれぞれのボタンを押せば簡単に作成できる。

最後に列幅や高さの調整、セルの書式の設定、タイトルの配置や配色などを設定すれば完了である(画面 13)。

Step 3

アプリケーションの実行

Excel をデザインしたあとに、ユーザーに Excel を実行してもらおう。ユーザーはセルへの条件や値の入力と、抽

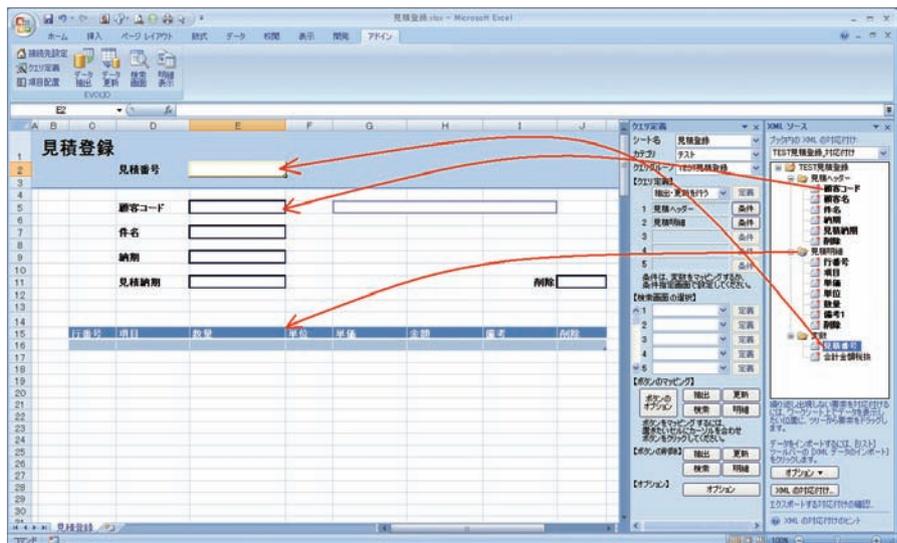
出・更新ボタンを操作してアプリケーションを実行する(画面 14)。

ここでデータを更新する最も基本的な流れは、抽出条件をセット→「抽出」ボタン→抽出されたデータに対して追加・更新・削除データをセット→「更新」ボタン、となる。

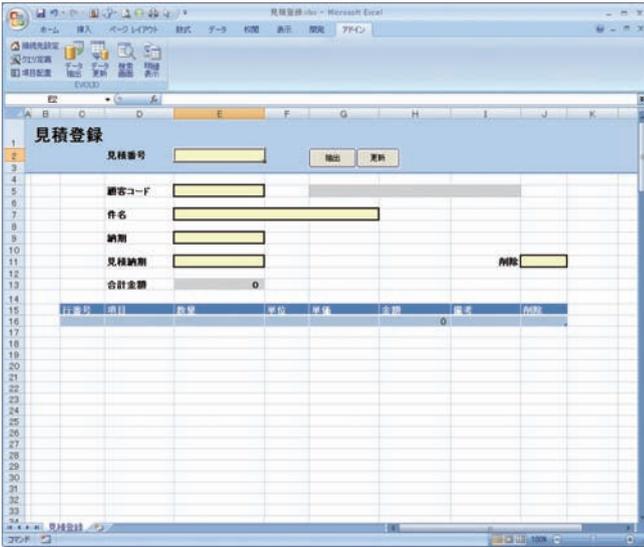
ポイントは、追加・変更・削除の項目やレコードが何件あっても、データをセットしたあとに「更新」ボタンを 1 回押すだけで実行するところにある。更新のためのデータセット方法は、以下の通りである。

- 1 新規見積は、見積番号に新規コードをセット
- 2 見積明細の追加は複数表示行の最下行に追記
- 3 内容変更は対象項目を上書き
- 4 見積明細の削除はその行の削除列に数字の 1 をセット
- 5 見積の削除は見積ヘッダーの削除項目に数字の 1 をセット

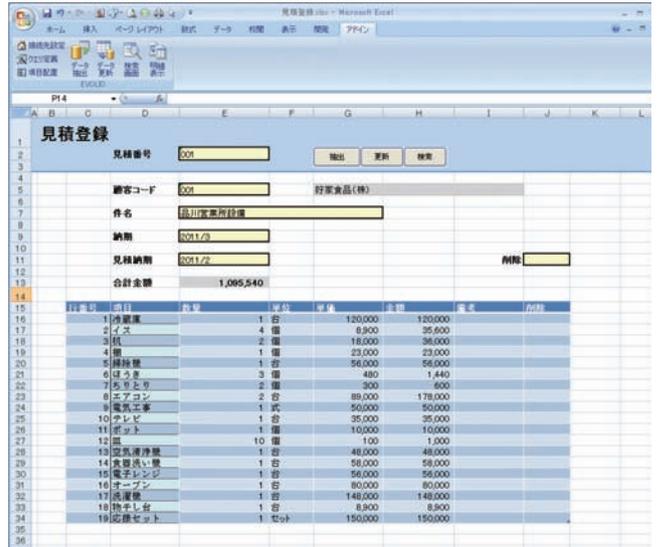
画面 12 項目配置



画面13 Excelのデザイン完了



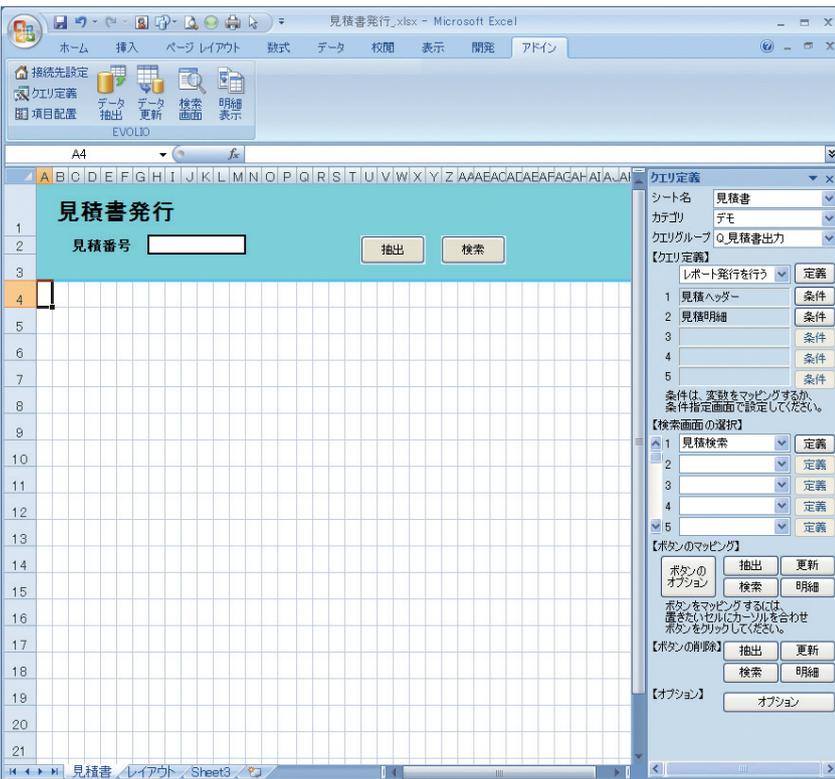
画面14 「見積登録」の実行イメージ



オペレーションイメージを知る ②

IBM i のDBからデータを抽出してレポートを発行する例

画面15 見積書発行の実行用シートとクエリ定義



次に、レポート発行の例を説明しよう。EVOLIOのレポート機能は、OPEN XMLというテクノロジーを使用しており、データをセルに貼り付けるようなオブジェクト操作に代わり、XMLデータをファイルに書き出すことで高速に動作する。

レポートの作成手順は、DBのデータを抽出・更新するアプリケーションと同じで、「データセンターでのDB連携定義」してから「Excelのデザイン」を行う手順だ。実際に見てみよう。

Step 1

レポート発行のDB連携を定義

データセンターでのDB連携定義

は、「見積登録」で作成したクエリーとほぼ同じであるため手順は割愛する。レポート発行の場合はDB更新しないので、クエリー全体のプロパティ定義で、更新処理の部分を「更新なし」にすればよい。

Step 2

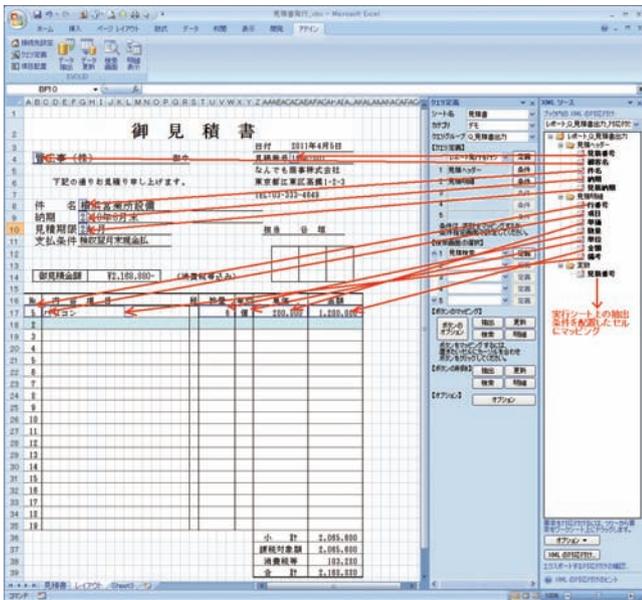
Excelのレポートをデザイン

レポートの場合は通常、抽出ボタン

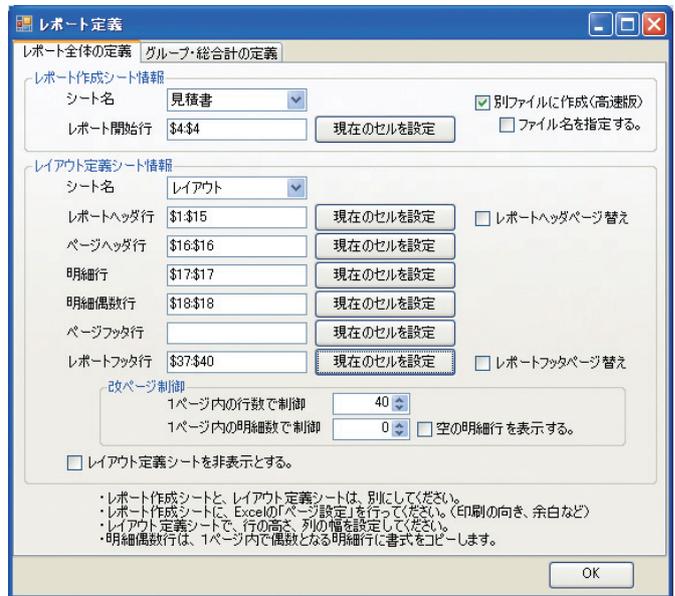
を配置し実行結果を表示するための実行用シートと、レポートのレイアウトを定義するレイアウト定義シートの2つを使用する。

最初にクエリー定義ペイン上で見積書出力のクエリーグループを指定後

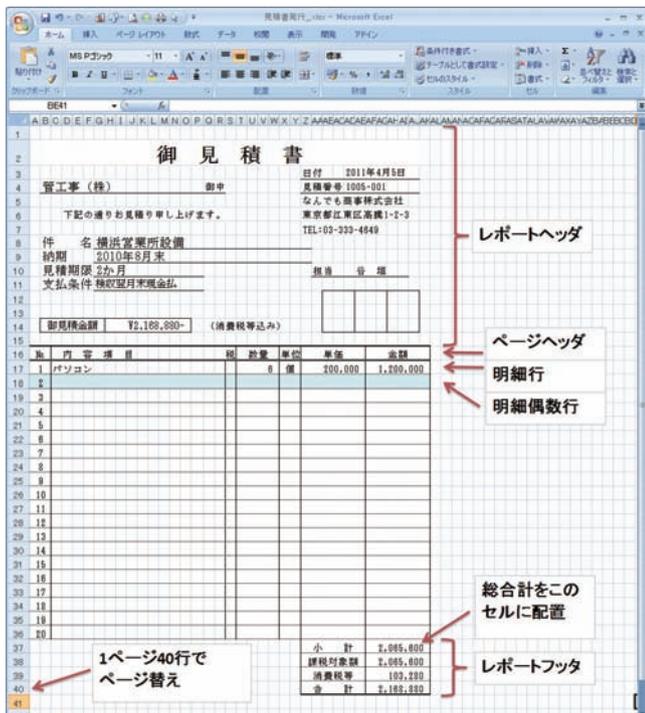
画面 16 見積書のレイアウトに項目配置する



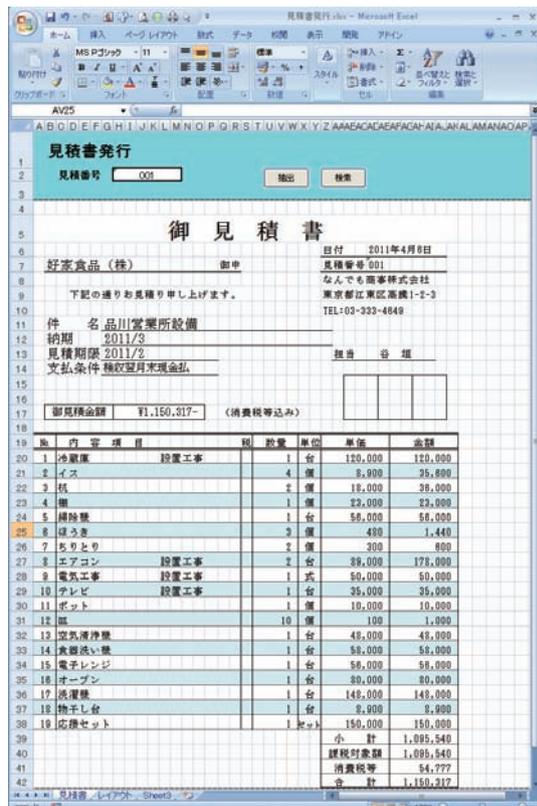
画面 17 レポートの定義ウィンドウ



画面 18 レイアウトの定義



画面 19 Excelで見積書を抽出したイメージ



に、「レポート発行を行う」を選択する(画面15)。

次に、レイアウト定義シート上に見積書のレイアウトを作成し、項目を配置する。項目配置は、抽出条件に対しては実行シート上にマッピングし、レポート出力項目のマッピングはこのレイアウト定義シート上に対して行う(画面16)。

そして、クエリー定義画面のレポート発行定義ボタンをクリックしてレポート定義ウィンドウを呼び出す(画面17)。ここでは、レポートを出力するための「レイアウト全体の定義」、キープレークでのページ替えや小計の指定などを行う「グループ・総合計の定義」を行う。

「レイアウト全体の定義」では、レ

ポートの最初にのみ印字(表示)するレポートヘッダー行、ページの最初にのみ印字(表示)するページヘッダー行、レポートの最後に印字(表示)するレポートフッター行、ページの最後に印字(表示)するページフッター行の位置と、データの各行ごとに印字(表示)する明細行の開始位置を設定する。それぞれ複数行あっても構わない。また、偶数行は背景色を変えたい場合には明細偶数行を別途指定する。

ここでは、画面18のレイアウトに合わせてそれぞれの値を設定した。ページ替えはExcelのセル行数で40行ごととした。この見積書の例ではキープレークはないが、全体の総合計として金額の合計を設定し、合計を表示するセルを指定した。

以上でレポートの定義は完了したので、Excelを保存してアプリケーションを実行する。

Step 3

レポート発行 アプリケーションの実行

画面19はレポート発行アプリケーションの実行イメージである。メインのシートで見積番号を検索し、指定後に抽出ボタンをクリック。見積書にデータを配置して表示する。プレビュー機能で改ページを確認できる。

EVOLIOが備える多彩な機能を知る

RPGとの連携で本格的な業務アプリケーション

ここまで簡単にアプリケーションの作成方法を説明したが、これ以外にもEVOLIOは多彩な機能を備えている。それらの組み合わせで、本格的な業務アプリケーションの構築が可能になる。

Function 1

RPG/CLプログラムの 呼び出し

ExcelからWebサービス経由で、IBM iのILE形式のCLまたはRPGプログラムを呼び出せる。画面上の入力項目に対する複雑な入力チェックや、

画面20 RPGを呼び出すクエリーの作成



複数のテーブルを更新するアプリケーションの作成には、RPGなどで処理を

記述する。RPGプログラムの呼び出しは、繰り返し明細行であっても1行

ずつパラメータを渡して呼び出せる。
 例えば、Excel上の明細行に開始日と終了日という2つの列がある際、その大小チェックをRPGで実行する場合、**リスト1**のようなプログラムを用意する。

データセンターでのDB連携定義で、更新時にDBを実更新する前に、このRPGプログラムを呼び出すクエリーを定義する(画面20)。あとはExcelにこのクエリーを含むクエリーグループを定義すればよい。**画面21**はその実行例である。

特に複雑な処理の場合は既存のビジ

ネスロジックを流用したり、RPG開発者との作業分担も可能である。

Function 2

検索画面

EVOLIOでは、データ入力時にマスターのコードがわからない場合などを考え、検索画面を自動的にポップアップできる。データセンターで「検索クエリー」という簡単なクエリーを

作成しておけば、Excel側でシートごとに検索クエリーを最大10個まで定義できる。(画面22)

Function 3

その他の多彩な機能

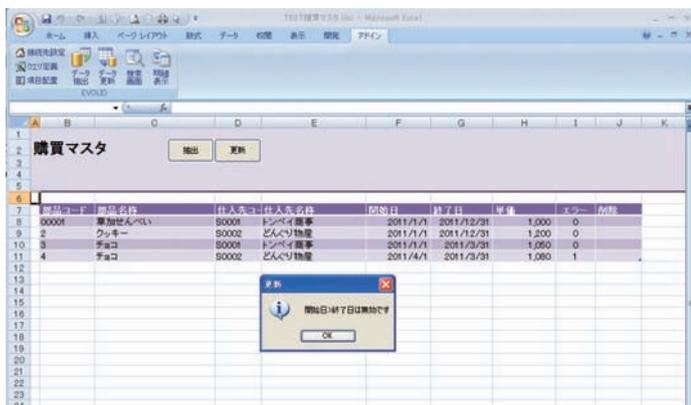
- 1 CLやRPGプログラムと同様に、DB内のストアードプロシージャを呼び出せるので、開発効率を高められる。

リスト1 大小チェックを行う RPG プログラム

```

C      *ENTRY          PLIST
C
C      PARM                      STRDTE          8 0
C
C      PARM                      ENDDTE          8 0
C
C      PARM                      OUTRTN          1 0
C
C      STRDTE          IFGT          ENDDTE
C
C      Z-ADD          1                      OUTRTN
C
C      ELSE
C
C      Z-ADD          *ZERO          OUTRTN
C
C      END
C
C      SETON                      LR
C
C      RETURN
  
```

画面21 RPG を呼び出す実行結果のイメージ



画面22 検索画面の例



- 2 SQL文を直接記述して、クエリーにできる。再帰呼び出しを使用するような複雑な抽出時に向く(例: 生産管理で使用する部品構成表の複数階層の抽出など)。
- 3 ドリルダウン用に明細ボタンがあり、1シート目で表示中の特定行に対する明細を別シート表示させるようなアプリケーションを作成できる(例: 日次でサマリーした実際データを検索、チェックしたい行を選択して明細ボタンを押し、その行の明細データを参照)。
- 4 Excel上の特定のセルにコード値を入力した時に、DBからコード値を検索して別のセルに名称を自動表示。
- 5 テーブル作成機能があるので、コーディングの知識がなくても、小規模システムを新規構築する場合などに便利。
- 6 Excelを開く時、ユーザー/パスワードでログインできる。カテゴリごとに、ユーザーとロールに権限(参照/更新/定義)を与え管理できるので、DBスキーマに対しても実行可能。
- 7 ユーザーのログイン履歴や抽出・更新の操作ログを記録し、ある権限者によってそのログを参照管理できる。
- 8 Excelのバージョンを管理できるので、最新バージョン以外のExcelファイルを使用させないなどの制限をかけられる。6と7も併せ、セキュリティを確保できる。
- 9 Excelのピボットテーブルやグラフ作成機能と組み合わせられる。EVOLIOでDBからのデータ抽出時に使用したり、3のドリルダウンと組み合わせれば、簡易BIツールとしての利用が可能。

C O L U M N

排他制御とEVOLIO

「楽観的排他」と「悲観的排他」のすみ分け

ここで、排他制御に少し触れておきたい。排他制御には、「楽観的排他」と「悲観的排他」の2種類がある。業務内容にもよるが、ここでは楽観的排他制御を推奨したい。これは、抽出時は特に排他処理を行わず、更新時に他から更新されたか否かを確認し、もし他から更新されていたら自らの更新処理を破棄してエラーにする手法である。

EVOLIOではデータセンターで、「更新カウンタ」を非表示の項目として作成し、チェック用のRPGを使って2番目の更新時にはエラーを表示させる(リスト2)。

一方の悲観的排他制御は、他の処理との競合が認められないトランザクションで、開始時に更新が抑止されていないことを確認したあと、他からの更新

を抑止し、更新する際に抑止情報を解除する。例えば、「チェックアウトフラグ」を作成し、更新制御することも可能。

ただしEVOLIOの特徴を活かせる大量データを一度に更新するような使用方法では、多くのデータの更新を抑止してしまう点で、悲観的排他制御はあまり向いていない。従来のアプリケーションとの使い分けが必要だろう。

まずは、「排他制御は必要ない」あるいは「楽天的排他制御で運用可能である」場合は、EVOLIOを使うというのが、ざっくりとした目安である。排他制御の有無については、1つの更新処理の中で、EVOLIOで抽出をするデータの範囲を絞り、入力者の分担範囲と合わせるように設計することがポイントになる。

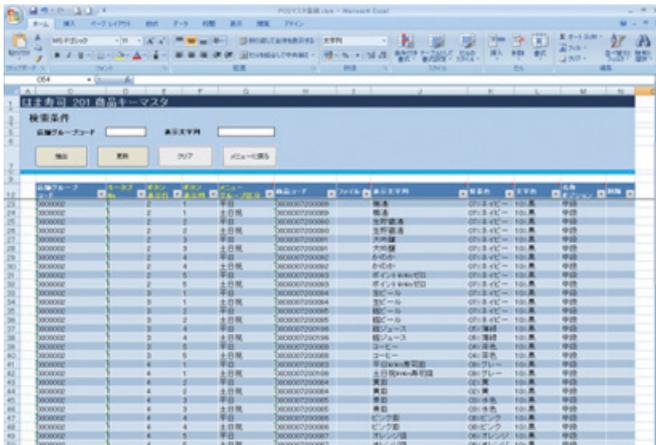
リスト2 更新カウンタチェック用RPGの例

```

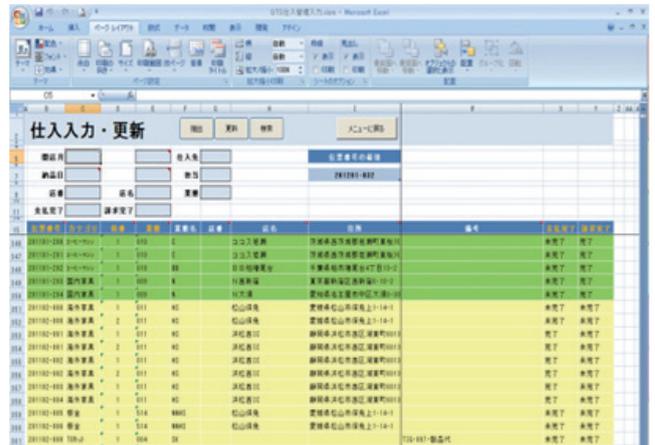
H*****
H*   顧客マスター更新カウンタチェック
H*****
C   *ENTRY          PLIST
C           PARM                CUSTCD          5
C           PARM                INCTR           10 0
C           PARM                OUTRTN          1 0
C           PARM                OUTMSG          100
C*****
C*   メイン処理
C*****
C*   顧客コードが顧客マスターに存在しない時は追加
C*   とみなしOKで返す
C           Z-ADD      *ZERO          UPDCTR          10 0
C/EXEC SQL
C+ SELECT UPDCTR INTO :UPDCTR
C+ FROM ZTEST/CUSTM
C+ WHERE CUSTCD = :CUSTCD
C/END-EXEC
C   SQLCOD          IFNE      *ZERO
C           Z-ADD      *ZERO          OUTRTN
C           ELSE
C*   更新カウンタが一致しなければエラー
C   UPDCTR          IFEQ      INCTR
C           Z-ADD      *ZERO          OUTRTN
C           ELSE
C           Z-ADD      1              OUTRTN
C           MOVEL      '排他エラー'  OUTMSG
C           END
C           END
C           SETON
C           RETURN
    
```

LR

画面 23 POS マスター登録システム



画面 24 仕入販売管理システム



実際の開発例を知る

ゼンショーグループで運用されているシステム

ここまでで、EVOLIOによる開発の概要を簡単に解説した。RPG連携や高速レポート発行などもサポートし、単にデータを抽出・分析するのではなく、業務アプリケーションを迅速かつ簡単に作成するための開発ツールであるとおわかりいただけたらと思う。

最後に、EVOLIOを利用する際の注意点として、以下の3点を挙げておく。

- (1) クライアントマシンのハードウェア性能
- (2) クライアントマシンのOSやExcel等のバージョンおよびファイル拡張子の統一
- (3) EVOLIOで作成したExcelファイルの配布作業

以下にゼンショーでの実際の開発例をご紹介します（カッコ内は、構築人日を示す）。

●POSマスター登録システム－マスター登録に活用－(33人日)

店舗POSで使用する20種のマスターを、各項目の論理チェックや整合性チェックなしで単純に表形式で登録させる。すべてのマスターを登録したあと、整合性チェックを実行し、エラーや警告メッセージを表示して次のアクションにつなげる(画面23)。

●仕入販売管理システム－既存のExcel販売台帳をDB化－(35人日)

テーブルウェアを販売している子会社で、商品マスターへ登録せずに販売する物品(主にカタログ品)については従来、Excelの台帳で管理していた。この台帳をDB化し、請求書発行済みか否かを管理することで、基幹システムと連携させた(画面24)。

●商品別・時間別販売速報－既存DBを活かし集計表示－(2人日)

POSからの速報データはDB上に

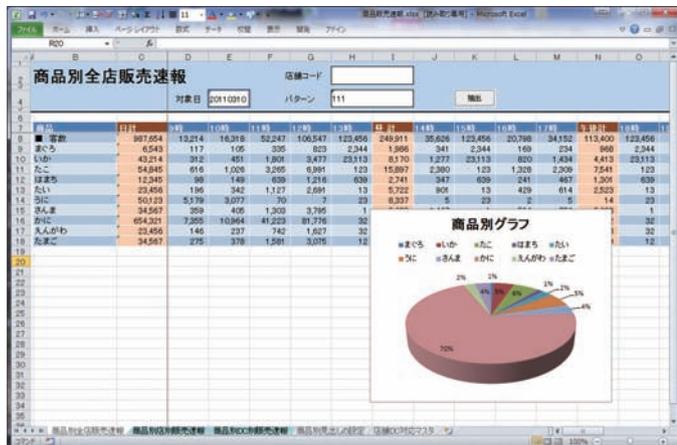
あったが、POS上の商品キーを自由にグルーピングして毎時間ごとの販売個数を集計して表示する機能を自由度を保ったまま整備。すき家などの新商品の売上傾向をリアルタイムで参照し、食材手配に活かす(画面25)。

●物流システム－アウトプット帳票の作成－(43帳票120人日)

物流システムのDBから、ディスプレイセンターのピックアップリストを出力(画面26)。

基幹業務の多様な周辺システムやサブシステム、情報系に近いシステムなど。またレガシーシステムを改善したいがその工数が大きすぎるケース、期間限定アプリケーション開発のケースなど、従来の開発手法では、投資結果が出なかったロングテール領域のシステムが企業内には多数存在する。EVOLIOは、これらのロングテール領域まで拡大して利用できるだろう。

画面25 商品別・時間別販売速報



画面26 物流システム

The screenshot shows an Excel spreadsheet with columns for item codes, item names, and dates. The table is titled 'アウトプット集計表' and contains data for various items and dates, including a date '2010年12月7日'.