

コンポーネントプログラミング

ポリテクセンター京都
(京都職業能力開発促進センター)

秋田 正秀

1. はじめに

最近のソフトウェアの雑誌や本を見るとコンポーネントを扱ったものが多く見られます。例えば、J2EEやActiveX関連などがそうです^{1) 2)}。ソフトウェア開発の現場では、コストを抑えながら納期を短縮するためにソフトウェアのすべてを自社開発から、コンポーネントソフトウェアを購入しそれらをつなぎ合わせてソフトウェアを開発する形式に変化してきていることがいわれています³⁾。コンポーネントの代表的なものには、Visual Basicで使われているActiveXコントロールやオートメーションオブジェクト、Javaで使われているJava BeansやEnterprise Java Beansなどがあります。

一方、コンポーネントの利用形態の面から見ると、ビジネスロジックといわれるものが増えてきています。ビジネスロジックは、簡単にはコードのコンポーネントで視覚的要素を持たないものとして作成されるものです。代表的な例としては、オートメーションオブジェクトやEnterprise Java Beansがこれに該当します。

本内容では、このコンポーネント開発としてセミナーのモデルカリキュラムやアビリティのユニットでも取り上げられているCOMコンポーネントについて紹介します。

2. COMの概要

COMは、Microsoft社が策定したコンポーネント間の通信規約です。この技術は、Microsoft社の中心的な技術の1つで、OLE (Object Linking and Embedding) から端を発しています。例えば、ActiveXコントロールやオートメーションオブジェクトなどの基礎になっています。

COMは、裏方的な役割を持ちコンポーネントが持つインターフェイスを公開します。COMコンポーネント必須の条件に、IUnknownインターフェイスを実装するという内容があります。IUnknownインターフェイスは、C++言語の抽象クラスとして作成されており、そのなかではQueryInterface、AddRef、Releaseという3つのメソッドが定義されています。これらの内容は、COMの仕様で決められており、その仕様に従って実装することになっています。

QueryInterfaceは、インターフェイスを公開するメソッドです。COMコンポーネントを利用するプログラムが、コンポーネントにインターフェイスの問い合わせをする際の窓口になるメソッドです。

AddRefとReleaseは、コンポーネントの寿命を管理するメソッドです。簡単には、コンポーネントの寿命を管理する変数があり、AddRefメソッドが実行されるたびに変数の値がプラス1され、Releaseメソッドが実行されるたびに変数がマイナス1されます。最終的に、変数の値が0になるとコンポーネントは消滅します。

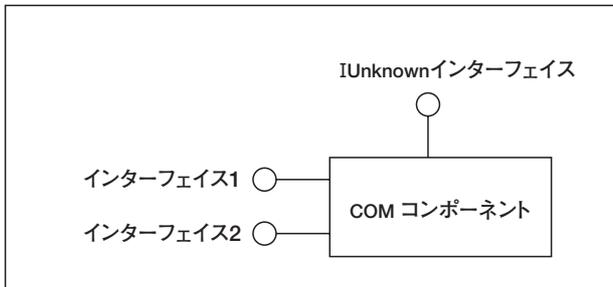


図1 COMコンポーネント

COMコンポーネントを図で表す場合、図1に示すように表されるのが一般的です。

COMコンポーネントは、矩形で表され、COMコンポーネントから伸びている手が、インターフェイスを表しています。COMコンポーネントでは、IUnknownインターフェイスは必須になっており、必ず上の所に伸びています。

また、COMは、バイナリ互換と位置透過性という特徴を持っています。

バイナリ互換は、言語に依存しないことを意味します。とにかく、COMに対応したものであればどんな開発言語でもよく、COMコンポーネントを作成した言語とCOMコンポーネントを利用する言語は、異なってもかまいません。例えば、Visual C++で作成したCOMコンポーネントをExcelやAccessで使うことができます。加えて、COMコンポーネントを利用するプログラムを作るときに、COMコンポーネントのソースコードを必要としません。インターフェイスの定義部分だけがあればよいことになっており、インターフェイスと実装を分離したことを意味しています。とりわけ、インターフェイスさえ変更されなければ、COMコンポーネントの実装が変更されてもCOMコンポーネントを利用するプログラムに影響はありません。

位置透過性は、COMコンポーネントを利用するプログラムを作成するときに、COMコンポーネントのファイルの物理的な位置を意識する必要がないということです。通常、ライブラリを使ってプログラムを作成する場合、C言語などでいえば、ヘッダやライブラリファイルの場所を特定しないと作成するプ

ログラムをコンパイルできません。要するに、ファイルの物理的な位置情報が必要であるということです。COMコンポーネントでは、基本的にそのような情報が必要なくなっています。必要な情報は、Windowsのレジストリデータベースから取得します。このような状況から、現在のところCOMに対応したOSは、Windowsしかありません。

3. IDispatchインターフェイス

Visual BasicやASPのプログラムでよく利用されているIDispatchインターフェイスを紹介します。IDispatchインターフェイスは、オートメーションを実現するインターフェイスです。IDispatchインターフェイスを使うと実行時にコンポーネントが持っているメソッドやプロパティを知ることができるようになります。通常、このような解決方法をレイトバインディングと呼んでいます。

例えば、次に示すVisual Basicのプログラムがあったとします。

```
Dim x as Object
Set x = CreateObject("Excel.Sheet")
x.Application.Visible = True
```

ここでのオブジェクト変数xの実体については、実行時に判明します。コンパイルの段階では、xが何であるかは解釈されません。

一方、次のようなプログラムはどうでしょうか。

```
Dim x as Worksheet
set x = Worksheets("Sheet1")
```

変数xは、特定のオブジェクトとして宣言されており実行前に解決されます。

前者のレイトバインディングを実現するのがIDispatchインターフェイスです。Visual Basicからコードコンポーネントを利用したり、ASPからコンポーネントを利用する際に使われています。

4. COMコンポーネントの開発環境

COMコンポーネントを作成するための開発環境にはVisual C++で使われているMFC (Microsoft Foundation Classes Library) やATL (Active Template Library), Visual BasicのActiveX DLL, ActiveX EXEプロジェクトがあります。

MFCでは、CCmdTargetクラスにCOMの機能が入っています。CCmdTargetクラスには、IUnknownインターフェイスが実装されており、このCCmdTargetクラスを継承して作成します。

ATLは、C++言語のテンプレート機能を利用した開発環境です。ATLでは、IUnknownインターフェイスを実装したテンプレートを用意しています。

Visual BasicのActiveX DLLプロジェクトやActiveX EXEプロジェクトは、Visual Basicのクラスモジュールを作成することでCOMコンポーネントができるようになっています。MFCやATLから考えると非常に簡単です。

5. COMの作成

ここでは、Visual C++ 6.0とATL3.0を使ったCOMコンポーネントの作成を紹介します。ATLでのCOMコンポーネントを作成する手順は、次のようになっています。

- (1) 「ATL COM AppWizard」の選択
- (2) 「ATLオブジェクトウィザード」の選択
- (3) インターフェイスの追加
- (4) メソッドの実装

5.1 ATL COM AppWizardの選択

Visual C++ 6.0のプロジェクト選択で、ATL COM AppWizardを選択すると図2に示すダイアログが表示されます。

この画面では、「DLL」や「EXE」を選択したり、MFCのサポートをできるようにしたりする選択ができます。ここでは、DLLを選択し、その他はデフォ

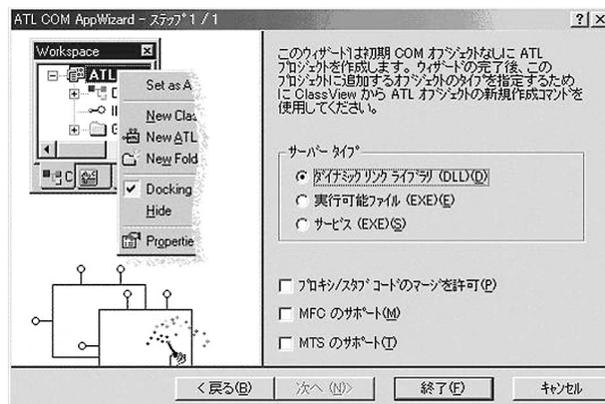


図2 ATL COM AppWizard

ルトのままとします。

5.2 ATLオブジェクトウィザードの選択

ATL COM WizardではCOMコンポーネントを入れる容器を決めたことになり、ATLオブジェクトウィザードではCOMコンポーネントの骨組みを決めます。図3にATLオブジェクトウィザードの画面を示します。



図3 ATLオブジェクトウィザード

コンポーネントのタイプの選択肢にはいくつかありますが、基本的な違いはインプリメントされるインターフェイスです。単純なCOMコンポーネントやオートメーションオブジェクトではシンプルオブジェクトが選択され、ActiveX コントロールではフルコントロールが選択されます。ここからも、ActiveXコントロールのベースの技術にCOMが使われていることが伺えます。

ここでは、シンプルオブジェクトを選択しウィザードを続けることとします。すると図4に示す画面

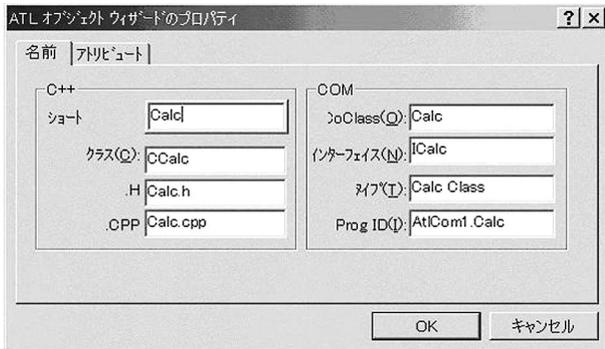


図4 名前

が表示されます。この画面では、COMコンポーネントの詳細を決めます。

名前タブの画面では、COMコンポーネントの名前となるCoClassやインターフェイスの名前、プログラムの中でクラス名として使われるProgIDやC++言語のクラス名やヘッダファイル名、実装ファイル名を定義します。

次の図5にアトリビュートタブの画面を示します。アトリビュートの画面では、スレッドモデルやインターフェイス、アグレーション（集成）の選択があります。スレッドモデルは、内容が難しすぎるのでここでは省略します。インターフェイスでは、デュアルとカスタムの選択があります。実行時バインディングのオートメーションをサポートするだけであればカスタムを選択し、アーリーバインディングもサポートするのであればデュアルを選択します。推奨は、デュアルです。

アグレーションでは、他のCOMコンポーネントの部品として機能できるようにするかしないかを選択します。

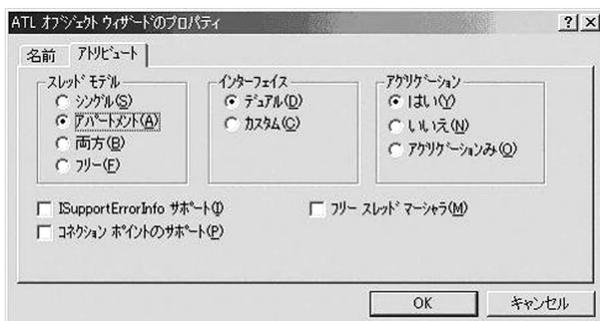


図5 アトリビュート

その他、主な選択としてISupportErrorInfoサポートとコネクションポイントのサポートがあります。ISupportErrorInfoサポートは、実行時のエラー処理を行う機能を提供するインターフェイスです。

コネクションポイントのサポートは、イベントのサポートで、COMコンポーネントに独自のイベントを発生できるようにする機能です。これにチェックを入れてイベントを実装すると、このCOMコンポーネントはイベントを発生できるようになります。これによって、このCOMコンポーネントを利用するプログラム側で、イベント処理プログラムを記述できるようになります。

ここでは、名前タブでショートのカラムにCalcを入力し、アトリビュートタブではISupportErrorInfoサポートにチェックを入れてCOMコンポーネントを作成します。

5.3 インターフェイスの追加

作成した結果のファイル一覧を図6に示します。

ATLオブジェクトウィザードを使ってCOMコンポーネントをプロジェクトに追加するとヘッダファイル（～.h）と実装ファイル（～.cpp）と合わせてインターフェイス記述ファイル（～.idl）が作成されます。COMコンポーネントでは、バイナリ互換を実



図6 ファイル一覧

現するためにインターフェイスの定義と実装が分離されているので、インターフェイス定義用のファイルが増えます。Visual C++でMFCを使ってActiveXコントロールを作成した経験がある方であれば、ODL (Object Definition Language) を思い浮かべてもらえばよいと思います。ATLでは、より一般的にインターフェイスの記述としてIDL (Interface Definition Language) を採用しています。IDLは、インターフェイス記述言語と呼ばれており、IDLの文法に従って記述されます。文法は、C++言語に類似しています。ただ、Visual C++を使ってATLでCOMコンポーネントを開発している場合は、そのIDLファイルを編集することは、そんなにありません。

クラスの一覧を図7に示します。図7より、インターフェイスとして、ICalcが表示されています。また、ICalcインターフェイスは、独立したICalcとCCalcクラスの中と2つ表示されていることがわかります。独立したほうが、IDLファイルに記述されている内容を表し、CCalcクラスの中のほうは、ヘッダファイルや実装ファイルでの実装を表します。



図7 クラス一覧

そして、公開する独自のプロパティやメソッドを追加します。例えば、このCOMコンポーネントは四則演算を提供するコンポーネントとします。このために、メソッドとして足し算 (Add)、引き算 (Sub)、掛け算 (Mul)、割り算 (Div) を作ります。

次の図8にインターフェイスメソッドの追加画面を示します。この画面では、メソッド名とパラメー

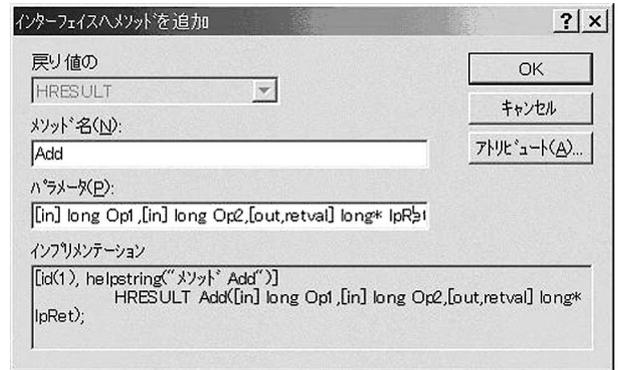


図8 インターフェイスメソッドの追加

タを記述します。インターフェイスの記述になるのでIDLの文法に従って記述されます。

メソッド名は、公開されるインターフェイスメソッドの名前を入力します。パラメータの部分で、IDLの文法が現れます。ここで記述している内容は、メソッドの引数として使うOp1とOp2、戻り値として使われるlpRetを定義しています。

インターフェイスメソッド定義後のクラス一覧を図9に示します。

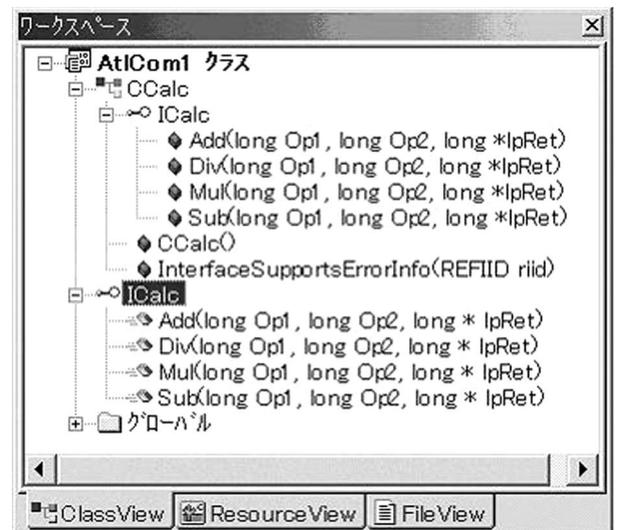


図9 インターフェイスメソッド

5.4 メソッドの実装

インターフェイスの定義ができると実装するメソッドのブロックができ上がっているのので、実装内容を追加します。ここでは、割り算の例を以下に示します。

リスト 1

```
STDMETHODIMP CCalc::Div(long Op1, long
Op2, long *lpRet)
{
    if (Op2 == 0)
        return Error("ゼロ割り算しています");
    *lpRet = Op1 / Op2;
    return S_OK;
}
```

プログラムは、分母が0の場合があるので、その場合は実行時のエラーを発生させています。

6. COMの利用

作成したCOMコンポーネントを利用したプログラムをVisual Basicで作成します。図10にVisual Basicのフォームを示します。数値1と数値2に数値を入力して→のボタンをクリックするとCalcコンポーネントが計算をして結果を返し、結果に表示するようになっていきます。



図10 クライアント

次にプログラムを示します。Calcコンポーネントは、インターフェイスをデュアルにしたのでアーリーバインディングをサポートしているのでその形式で記述しています。

リスト 2

```
Dim itf As New ATLCOM1Lib.Calc

Private Sub Command1_Click ()
    Dim lRet As Long

    On Error GoTo ErrorHandler1

    If Text1.Text = "" Or Text2.Text = "" Then
        MsgBox "数値を入力してください。"
        Exit Sub
    End If

    If IsNumeric(Text1.Text) = False Or
    IsNumeric(Text2.Text) = False Then
        MsgBox "数値を入力してください。"
        Exit Sub
    End If

    lRet = itf.Div(CLng(Text1.Text), _
        CLng(Text2.Text))

    Text3.Text = lRet
    Exit Sub
ErrorHandler1:
    MsgBox Err.Description
End Sub
```

7. まとめ

本内容では、COMコンポーネントの概要と合わせてATLを使ったCOM作成について紹介しました。COMコンポーネントは、OLEの技術をもとにできたWindowsのコンポーネントの技術です。COMを理解することで、プログラムにおけるコードコンポーネントについて理解できます。

一方、COMコンポーネントは、アビリティのマネジメント情報システム科のシステムユニットのシステム設計（コンポーネント）で取り上げられています。アビリティのシステムユニットでは、COMコン

ポーネントの作成をVisual BasicのActiveX DLLプロジェクトを使って行っているのが、今回紹介した内容より簡単にできます。このアビリティでは、最終的にはASPを使ったWEB-DB連携プログラムの作成にあります。ASPだけでは、独立したビジネスロジックを実現できません。COMコンポーネントをVisual Basicで作成してそれをASPで利用するように考えれば、独立したビジネスロジックを実現することができ、結果として再利用可能なコンポーネントを実現することができます。

＜参考文献＞

- 1) 石原直樹訳：「特集1 はじめてのBluePrints&J2EEパターン」、『WEB+DB PRESS』, Vol. 8, 技術評論社.
- 2) 大澤文孝, 初音玲：「特集1 Webアプリケーションがあるじゃないか」、『ドットネットマガジン』, 2003 JAN., 翔泳社.
- 3) 荒井玲子：「特集2 反復型開発のプロジェクト管理」、『Software People』, 技術評論社.
- 4) 瀬尾佳隆：『Visual C++ 6.0によるCOMプログラミング』, オーム社.
- 5) 加藤佐一：『ATLによるCOMプログラミング入門』, 技術評論社.
- 6) 田中正造監訳：『ATL COMプログラミング』, 翔泳社.
- 7) 豊田孝監訳：『ATLプログラミング』, ソフトバンク.
- 8) QUIPU LLC訳：『ATLインターナル』, アスキー出版局.
- 9) 豊田孝監訳：『VB COM』, 翔泳社.
- 10) 豊田孝：『Visual C++プログラマのためのCOM入門』, 翔泳社.
- 11) 豊田孝：『COMアーキテクチャと実践スクリプティング』, 翔泳社.
- 12) 竹内里佳監訳：『COMによる分散アプリケーション開発技法』, 日経BPソフトプレス.
- 13) 長尾高弘訳：『INSIDE OLE 改訂新版』, アスキー出版局.
- 14) バウングローバル(株)訳：『Inside COM』, アスキー出版局.
- 15) 豊田孝監訳：『インサイド COM+ 基本編』, 日経BPソフトプレス.

職業訓練用教科書 改定発刊のご案内

●職業能力開発総合大学校 能力開発研究センター編
●厚生労働省認定

社団法人 雇用問題研究会

〒103-0012 東京都中央区日本橋堀留町1-5-11
TEL 03-5695-0780 FAX 03-5695-0837
URL=<http://www.koyoerc.or.jp>

電気工学概論

B5判・196ページ/定価1,260円(本体1,200円)

機械工作法

B5判・418ページ/定価2,100円(本体2,000円)

電気応用

A5判・208ページ/定価774円(本体738円)

電気製図

B5判・268ページ/定価1,890円(本体1,800円)

電子工学

B5判・336ページ/定価1,890円(本体1,800円)

電気材料

B5判・256ページ/定価1,680円(本体1,600円)

電気工事

B5判・392ページ/定価2,520円(本体2,400円)

電気理論

B5判・268ページ/定価1,680円(本体1,600円)

電気関係法規

A5判・290ページ/定価1,575円(本体1,500円)

木造建築実技教科書

A4判・124ページ/定価1,680円(本体1,600円)