インターフェースによるマルチドキュメント型アプリケーション

大西孝臣*

Multi-document application with interface

Takaomi OHNISHI

Abstract

This article shows the technique developing a multi-document application. This application consists of two independent applications, each of which has only a document type and has an IDispatch interface to be an automation component, and a GUI content, which is to be an automation client, written in one of the several GUI-based programming languages or scripts. The automation, which is a COM-based technology, allows the automation client content to bind two of the automation component applications at run time.

This approach is one of key technologies to developing the experimental environment for learning digital signal processing.

1. はじめに

本稿では、ディジタル信号処理の実習を行うために用意した2つのドキュメントタイプをそれぞれのグングルドキュメントのWindowsアプリケーションとして実現して、さらに各アプリケーションにインターフェースを実装させてコンポーネント化を施し、別途作成した教材コンテンツからコンポーネント化された各ドキュメントを呼び出させることを可能にさせた実習環境について、その開発の技術面を紹介する。

2. システムの概要

2.1 実習環境の構成

本実習環境においては、学習者は、図1に示すような、プログラミング言語を扱った実習環境を提供する「コンパイル環境」と、数値データ処理としてのFFT及び逆FFTの性質を学ぶことに重点を置いた「数値データ・グラフ表示環境」の、計2つのドキュメントタイプと実質的に向き合うこととなる。各ドキュメントタイプは、シングルドキュメントのまま独立して、インターフェースの実装によりコンポーネント化された、MDIアプリケーションとして実現されている。

図1 実習環境の構成

本稿では、これら2つのアプリケーションの名称を、開発時において、コンパイル環境について「component05」、数値データ・グラフ表示環境について「component06」とした。

* 助 手 情報工学科

図1の構成に従う形に実現した実習環境の概観を図2に示す。

図2における教材コンテンツは、HTMLを使用してWebブラウザに表示されており、Webスクリプト言語であるVBScriptによるフォームにおいて、適当な箇所にあるテキストボックスにファイル名を指定してボタンを押すことにより、アプリケーションcompile05に対するドキュメント
２．２ オートメーションとIDispatchインターフェース

オートメーションは、アプリケーションが実装している機能を、Visual BASICやVisual C++、WordやEXCELなどのVBA、Webスクリプト言語などによる外部のプログラムから利用できるようにする、COMの実用技術の１つである。アプリケーションが、外部に対して、自身のオブジェクトの機能をアクセス可能にすることにより、そのオブジェクトの自動化が行われたことになる。自動化が行われるオブジェクトを持つアプリケーションをオートメーションコンポーネントと呼び、コンポーネントと外部から呼び出すプログラムなどをオートメーションクライアントと呼ぶ。

オートメーションコンポーネントにおけるオブジェクトの自動化は、COM標準インターフェースであるIDispatchインターフェースの実装により実現される。

IDispatchインターフェースの原理について、図３に示す。

一般にIUnknownインタフェースから派生される、カスタムなCOM標準インターフェースを持つオブジェクトの場合は、オブジェクトが公開する関数はvtableにおいてそれらのポインタを示されるのに対し、IDispatchインターフェースは、vtableからアクセスできる関数を限定させ代わりに、vtableよりも操作性に自由度を持たせたディスパッチテーブルと呼ばれる、独自のIDと関数の外部名との対応表を別に用意して、バインドするオートメーションクライアントアプリケーションの視点で見ると、ディスパッチテーブルに載せられている関数が、事実上外部に公開されているという方式である。

ディスパッチテーブルに載せられているIDはディスパッチID（dispID）と呼び、関数はディスパッチ関数と呼ぶ。

詳細なディスパッチの公開は、コンポーネント化する2つのアプリケーションであるcompile05とcomponent06は、同じ手続きにより、図４に示すような、それぞれ同じディスパッチ関数を公開する。

図３ IDispatchインターフェースの原理
vtableからアクセスできるGetIDsOfNames関数とInvoke関数が主として、外部からディスパッチ関数へのアクセスの役割を担う。

図４ 実習環境において公開するディスパッチ関数
コンポーネントオブジェクトの機能は、FileNameというプロパティとShowApp、ShowWindow、LoadFileという3つのメソッドの4つ外部名で公開されている。FileNameプロパティに関するディスパッチ関数は取得関数のGetFileNameと設定関数のSetFileNameであり、3つの公開メソッドに関するディスパッチ関数の
2.4 ドキュメントのロード

オートメーションクライアントである教材コンテンツにおいて、オートメーションコンポーネントであるアプリケーションcompile05あるいはcomponent06を呼び出し、ドキュメントをロードさせる手続きについて、図5に示す。

図5 コンポーネントにおける
ドキュメントのロードの手続き

図5における手続きにおいて、コンポーネントに関する詳細を知らされていない指導者が教材コンテンツを作成するという状況を想定して、教材コンテンツにおけるコーディングの量を減らす目的から、クライアントがオンタイム時にディスパッチテーブルを参照するレイアウトィングクティブ方式を採用した。

教材コンテンツは、コンポーネントの起動とドキュメントのロードを指示するために、「component05.document」といったプログラマティックID（progID）と呼ばれる文字列を用いる。このprogIDによるクライアントからの指示は、レジストリにおいて、コンポーネントオブジェクトが持つ固有のIDであるクラスID（CLSID："ED9C0CAD-7730-4EA4-9B9D-1BC345FE58B5"といった128bitのバイナリデータ）、そしてアプリケーションの実行ファイルへと参照されて、component05などのコンポーネントアプリケーションが実行される。続いて、Invoke関数の働きによって、ディスパッチ関数LoadFileが実行され、ドキュメントファイルがコンポーネントアプリケーションにロードされる。

2.5 教材コンテンツの実現

図2にて概要を示した、HTMLとVBScriptを使用した教材コンテンツにおいては、図6に示すように、フォーム、ボタンコントロール、テキストボックスコントロールを配置した。

図6 教材コンテンツにおける
フォームとコントロールの配置

教材コンテンツのソースリストの要点についてまとめたものを次に示す。
フォームが設定され、inputタグによって、ボタンコントロールとテキストボックスコントロールが配置されている。テキストボックスコントロールについてはinputタグのvalueオプションを使用することにより、ロードするドキュメントファイル名の初期値を与えることができる。

head部内のscript部において、Web用スクリプト言語としてVBScriptが選択され、各ボタンコントロールにおける左シングルクリックイベントに対するイベントドライプ用のサブルーチンが載せられている。

Visual BASIC、VBA、VBScriptにおいては、OpenObjectメソッドを用いることで、コンポーネントオブジェクトに対するインターフェースを確立する。上記のリストの場合、オートメーションコンポーネントのprogIDを用いてシングルクリックを呼び出しているため、OpenObjectメソッドの戻り値はIDispatchインターフェースのオブジェクトということになり、戻り値を代入された変数MyObject01あるいはMyObject02のデータ型はObject型となる。VBScriptにおけるObject型は、Javaにおけるjava.lang.Objectクラスに相当する。

クライアントからオートメーションコンポーネントのディスパッチ関数に対するアクセスは、MyObject01.ShowAppやMyObject02.FileNameといった具合に、“(IDispatchインターフェースのオブジェクト名) (ディスパッチ関数の外部名)”という形態で行う。本稿の例では、空のドキュメントを表示する場合に無いので、ディスパッチ関数ShowWindowは実行していない。

テキストボックスコントロールに書き込まれたファイル名の文字列は、CStr(Document.form01.FileName01.value)といった具合に獲得することができ、コンポーネントのFileNameプロパティに代入する。

3. おわりに

本稿では、オートメーションという手法を用いてアプリケーションにコンポーネント化を施し、ドキュメントを統合させるアプローチについて紹介した。

オートメーションを採用することにより、多くのプログラミング言語やスクリプト言語を用いての教材コンテンツの作成が可能となった。これまでに、本稿において紹介したVBScriptの他にもVisual BASICとEXCEL VBAを用いての教材コンテンツを実現している。結果、指導者に親しみやすく、コンテンツの作成に効率の良い言語を選択することができるようになった。

参考文献

1）大西孝臣、杉岡一郎共著、工業高等専攻科向けの数値処理の実習教材の開発、情報処理学会第63回（平成13年後期）全国大会講演論文集、vol.4、pp.141-142、2001

2）David J. Kruglinski, George Shepherd, Scot Wingo共著、(有)デジタルアドバンテージ訳、プログラミングMicrosoft Visual C++ 6.0、日界BPソフトプレス、1999

3）Julian Templeman著、㈱ランゲージジャーメンテーションサービス訳、パワープログラミングMFC COM、ソフトバンク、1998

4）Guy Eddon, Henry Eddon共著、農田孝監訳、インサイドCOM+ 基本編、日界BPソフトプレス、2000

5）MSDN Subscriptions Library日本語版、October 2001, Microsoft Corporation、2001

（平成13年11月30日受理）