

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別個のシ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の性質を学ばせる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かいあうこととなる。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。

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。

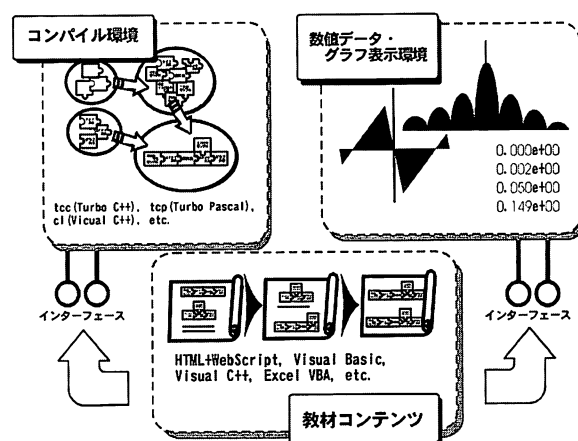


図1 実習環境の構成

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

名称は、外部名と同一である。

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component05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。

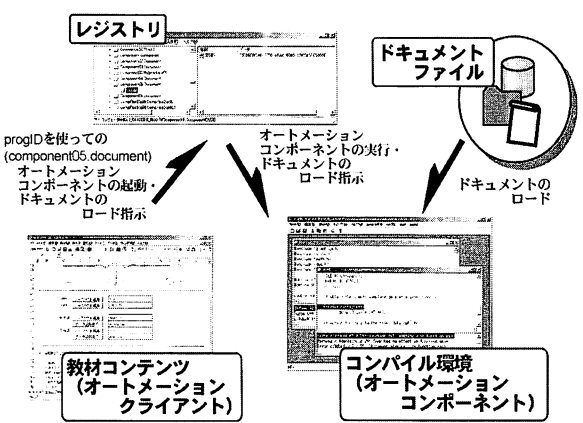


図5 コンポーネントにおけるドキュメントのロードの手続き

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トバインディングと呼ぶ方式を採用した。

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”などといったプログラマティックID (progID) と呼ぶ文字列を用いる。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にて概観を示した、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。

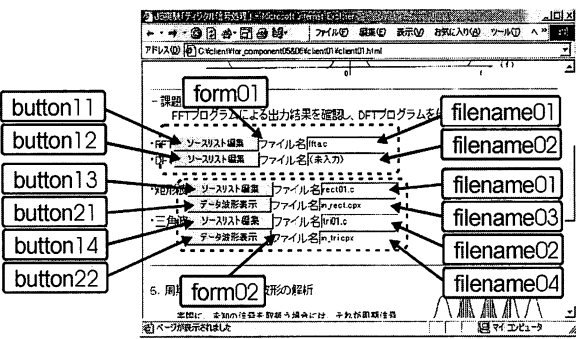


図6 教材コンテンツにおけるフォームとコントロールの配置

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。

```
<html>
<title>J 5実験「デジタル信号処理」</title>

<head>
<script language="VBScript">

Sub button11_OnClick
  Dim MyObject01
  Dim MyFileName
  Set MyObject01 = CreateObject("Component05.Document")
  MyObject01.ShowApp
  MyObject01.ShowWindow
  MyFileName = "C:\dsp¥" & CStr(Document.form01.filename01.value)
  MyObject01.FileName = MyFileName
  MyObject01.LoadFile
End Sub

Sub button12_OnClick
  ...
End Sub

Sub button13_OnClick
  ...
End Sub

Sub button14_OnClick
  ...
End Sub

Sub button21_OnClick
  Dim MyObject02
  Dim MyFileName
  Set MyObject02 = CreateObject("Component06.Document")
  MyObject02.ShowApp
  MyObject02.ShowWindow
  MyFileName = "C:\dsp¥" & CStr(Document.form02.filename03.value)
  MyObject02.FileName = MyFileName
  MyObject02.LoadFile
End Sub

Sub button22_OnClick
  ...
End Sub

</script>
</head>
```

```

<body bgcolor="#f7f7f7">
...
<form name=form01>
・FFT<input type=button name=button11 value="ソースリスト編集">
ファイル名<input type=textbox name=filename01 value="fft.c"><br>
・DFT<input type=button name=button12 value="ソースリスト編集">
ファイル名<input type=textbox name=filename02 value=" (未入力) ">
</form>

<form name=form02>
・矩形波<input type=button name=button13 value="ソースリスト編集">
ファイル名<input type=textbox name=filename01 value="rect01.c"><br>
<input type=button name=button21 value="データ波形表示">
ファイル名<input type=textbox name=filename03 value="in_rect.cpx"><br>
・三角波<input type=button name=button14 value="ソースリスト編集">
ファイル名<input type=textbox name=filename02 value="tri01.c"><br>
<input type=button name=button22 value="データ波形表示">
ファイル名<input type=textbox name=filename04 value="in_tri.cpx">
</form><br>
...
</body>
</html>

```

body部において、form01とform02の2つの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の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ブ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を用いてオブジェクト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面は無いので、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になった。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、工業高専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、日系BPソフトプレス、1999
- 3) Julian Templeman著、(株)ランゲージドキュ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、豊田 孝監訳、インサイドCOM+ 基本編、日経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）