

図3 実習教材の最小限のシステム構成

コンパイル環境、数値データ・グラフ表示環境の2つのユニットは、共に Visual C++6.0 を使って作成した、インターフェースのサーバ側の一形態であるオートメーション・コンポーネントである。サーバ・コンポーネントである2つのユニットそれぞれのインターフェースに、最小限の機能として、『ファイル名』プロパティと『ファイルのロード』メソッドとを実装させ、教材コンテンツからのアクセスを受け付けることにより、2つのユニットは学習の流れに沿って機能することができる。

ファイルがロードされたコンパイル環境において、学習者の指示によりコンパイルが行われる実行手順を図4に示す。

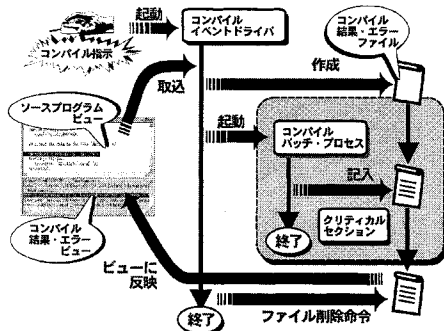


図4 「コンパイル環境」における実行手順

コンパイルの実行手順に関しては、昨年が我がが発表したもの<sup>1)</sup>と基本的に変化がないが、Windows Me 上にてコンパイルを実行する際に、バッチ処理に伴う標準入出力のリダイレクト処理に関連して、以前の Windows (98, NT4.0, 2000) においては起こらなかった現象が生じたために、対処が必要であることが判明している。

図2の例においては、コンパイル環境におけるコンパイラとして Turbo C++4.0 付属の tcc

コンパイラを組み込んでいるが、インターフェースに使用言語の選択に関する新たなメソッド・プロパティを実装させることにより、学習者自らが教材コンテンツに対する操作を通じて、使用する高級言語を指定できる。我々は他のコンパイラとして、Turbo Pascal 付属の tcp コンパイラ、Visual C++6.0 付属の cl コンパイラの組み込んだ動作確認をしている。

さらなる機能として、コンパイル環境が数値データ・グラフ表示環境に実装されたメソッドやプロパティにアクセスすることにより、例えば実習における課題の1つとして、コンパイル環境にて、数式のパラメータを少しずつ操作させて得た計算結果を逐一グラフで表示させるということも可能になる。

一方、教材コンテンツは、オートメーション・クライアントとして作成され、教科担当の指導者は、用意されたテンプレートを加工する形で教科に合わせたコンテンツを編集する。

図2の例においては、VBScript を使ってサーバ・コンポーネントのインターフェースにアクセスをしているが、その他にも、インターフェースへのアクセスが可能な教材コンテンツの提示手段として、VBScript 以外の Web 用スクリプト言語や Visual Basic、Visual C++、EXCEL VBA などを適用することができる。

さらなる例として、EXCEL VBA にて教材コンテンツを作成した場合、今度は教材コンテンツ側にインターフェースを実装させ、コンパイル環境がクライアントの立場になって教材コンテンツである EXCEL 側のインターフェースにアクセスすることにより、コンパイラの実行結果を教材コンテンツへ向けてフィードバックさせ、教材コンテンツ上に結果のグラフ表示をさせることも可能になる。

### 3. おわりに

本研究では、インターフェースを実装してコンポーネント化されたユニットで構成された実習教材を用いることによる、工業高専専攻科に向けた数値処理の実習を対象とした場合の有効性について提案した。現在は Linux 環境のユーザインターフェースやコンパイラとの関連付けについて検討している。

### 参考文献

- 1) 大西・杉岡「Windows アプリケーションにコマンドラインコンパイラを編入させた実習教材の開発」、教育工学関連学協会第6回全国大会、pp. 821-822 (2000)