

Visual Studio LightSwitch と SQL Server によるデータベースアプリケーション開発の概要

兵庫県立西宮香風高等学校主幹教諭
松本 吉生

1. LightSwitch とは

Microsoft Visual Studio LightSwitch は Windows アプリケーション開発ツールである Visual Studio のファミリー製品で、データベース連携の業務アプリケーションを素早く開発できることを目的に開発された。最初の製品は 2011 年 8 月に発売された LightSwitch 2011 で、現在は Visual Studio 2012 に統合された形で LightSwitch 2012 が提供されている。

Visual Studio は Windows アプリケーション開発における中心的アプリケーションとして長く開発が続けられてきた。近年では巨大アプリケーションを複数の開発者が共同で開発するための Team Foundation Server といったプロの開発者用の機能を充実させている。

一方で Visual Studio は、初心者を使いやすい環境づくりにも力を入れており、個人が学習のために使うには Express 版という無料のバージョンを用意するなどしている。Express 版の利用を想定した書籍も多数出版されており、学生などがプログラミングの学習をしやすくなっている。

無料の Visual Studio Express 版を使い、初心者向けの書籍を読みながらプログラミングをしてみると、誰でも簡単に電卓や住所録あるいは簡単なゲームなどの Windows フォームアプリケーションを作成することができるが、データベース連携のアプリケーションを作ることは簡単ではない。しかし業務アプリケーションの世界ではデータの一元管理が求められており、データベースアプリケーションを必要とする場合が一般的である。そこでデータベースアプリケーションを作ることに特化したツールとして LightSwitch が開発された。

2. LightSwitch が必要とされる背景

LightSwitch はデータベース連携アプリケーション

を素早く簡単に作ることを目的とした開発ツールだが、プログラミング初心者だけを対象としたものではなく、プロの開発者にとっても役立つものになっている。というのも、ひとつにはデータベースアプリケーションは、基本的にデータの問い合わせ、追加、更新、削除といったパターン化された機能を持つ必要があり、開発者はこのデータ接続に関する部分をコーディングする作業が定型化している実情があるからである。この定型作業を LightSwitch で自動化することにより、開発者はルーチンワークから解放され、データベースの論理構造の開発に集中できるメリットがある。もうひとつの理由として、比較的短期間で小規模のプログラミングを繰り返しながら開発を進める最近の開発スタイルに適していることがある。

典型的なソフトウェア開発プロジェクトではソフトウェア工学でいうウォーターフォール・モデルがとられる。ウォーターフォール・モデルでは、まずシステムで何を実現しなければならないかを要求定義し、実現する機能の論理的な外部設計を行い、そして実際のコンピュータシステムで実現するための内部設計をし、それから開発、テスト、運用などの作業工程に入るという一方向の流れで開発を行う。この開発手法では原則として前工程が確実に完了してから次の工程に進むこととされ、基本的に後戻りは許されない。前後工程の並行作業も行われない。

ウォーターフォール・モデルは、変更コストは開発が進むにつれ増大するという問題を避けるように考えられている。前工程の変更は後工程に大きな影響を与えるからだ。したがって最初に厳密な要求定義を行い、各工程のそれぞれで誤りがないことを確認しながら開発を行えば、最終的に正しい成果物が無駄なく合理的に得られるという考え方をとる。しかし実際には、とりわけ今日的な複雑で大規模なシステム開発において、このウォーターフォール・モ

デルがしばしば開発の失敗を引き起こす原因であると言われている。というのも、現実的に初期の段階で将来起きうる全ての問題を予測することはほぼ不可能であるためだ。またそもそも厳密な要求定義をすること自体が不可能である、という考え方もある。変化の激しい時代において、要求仕様を定義した後の何か月もかかる開発の間にビジネス環境が変わり、元の要求そのものが変化を余儀なくされることも多いからだ。

このウォーターフォール・モデルによる開発に対して「アジャイルソフトウェア開発」あるいは「エクストリーム・プログラミング」と呼ばれるモデルがある。これらの手法では開発を1～2週間単位の短い「イテレーション」と呼ぶ期間に区切り、その期間ごとに小さな機能を持つ中間完成品を生産してテストを行い、機能をチェックして要求、分析、設計を見直しながら進めていく。開発当初の要求仕様はおおまかなものにとどめておき、変化を当然のものとして受け入れながら柔軟に開発を進める。このような開発モデルにおいて、素早く一定の品質が確保されたアプリケーションを作成することができる開発ツールが求められており、LightSwitch もそのひとつである。

3. LightSwitch と OBA 開発

一般のデスクワーカーも今日ではコンピュータを活用したデータ処理能力が求められる。Excel に代表される表計算ソフトウェアの活用は当然であり、Access のようなデータベースソフトウェアの利用も一般的になりつつある。Excel では複雑な処理を自動化するために VBA 言語を使ったマクロプログラミングも活用される。デスクワーカーのこうしたソフトウェア活用は、それぞれに自分が何をしたいかが明確なデスクワーカー自身が問題解決のために必要なソフトウェアを選択し、自らカスタマイズして利用するところに特徴があり業務の効率化に寄与している。このような、いわゆるオフィスソフトを活用して業務システムに似たものを開発することを Office Business Application 開発、略して OBA 開発という。

LightSwitch は SQL Server に対してあたかも Excel や Access のような操作感で開発することができる。ロジックの組み込みは Excel の VBA を使

うよりわかりやすい。SQL Server と LightSwitch による簡単なシステム開発を OBA の一部に取り入れることは、データを一元管理できることやコードのメンテナンス性の面でオフィスワーカーにとっても価値が高い。

4. InfoPath と LightSwitch の違い

筆者はこの10年、勤務校で教務システムの開発と運用に携わる中で、データベースを SQL Server とし、クライアントに InfoPath や Access を利用する OBA 開発によるシステムに取り組んできた。このクライアントサーバー型のデータベースシステムにおいては InfoPath をクライアント側からのデータ入出力に利用するが、これを LightSwitch による開発に置き換えることができる。クライアント側に InfoPath を用いた場合と LightSwitch による開発の場合の違いをまとめると次のようになる。

・ InfoPath の場合

コンピュータに InfoPath のインストールが必要
フォームのデザインと確認が素早くできる
コードを書く必要がない

・ LightSwitch による開発の場合

作成したプログラムの実行環境に制約はない
開発上ビルドと配布の作業が必要になる
ロジックの実装はコードを書く必要がある

5. LightSwitch と SQL Server によるシステム

LightSwitch による開発では、デスクトップアプリケーションや Web アプリケーション、クラウド上に展開するアプリケーションを作ることができ、接続対象となるデータベースには SQL Server, SQL Azure, SharePoint などがある。ここではデータベースを SQL Server とし、一般のデスクトップアプリケーションを作成する概要を説明する。

6. SQL Server のテーブル作成

成績管理を念頭におき、生徒の情報を格納するテーブルを SQL Server に作成する。SQL Server Management Studio を使いフィールド名とデータ型を以下のように設定する。

フィールド名	データ型
学籍番号	nchar(7)
氏名	nvarchar(50)
性別	nchar(1)
生年月日	date
学年	int



Fig.1 「生徒」テーブルの作成

次に Visual Studio LightSwitch を起動し、新しいプロジェクトを開始して新しいソリューションを作成する。テンプレートは言語を Visual C# とし、LightSwitch アプリケーションを選択する。プロジェクトを開始すると次のようなデザイナーが表示される。



Fig.2 LightSwitch プロジェクトの開始画面

「データを使用して開始」で「新しいテーブルの作成」か「外部データソースにアタッチ」のどちらかを選択するが、ここでは既存の SQL Server に作成した「生徒」テーブルに接続するため「外部データソースにアタッチ」を選択する。すると「データソースのアタッチウィザード」が起動し、項目を選択しながらいくつかのステップでデータ接続を構成することができる。

ウィザードによってデータベースを SQL Server にし、先に作った「生徒」テーブルに接続すると、構成されたデータ接続がデザイナーに表示され、SQL Server の「生徒」テーブルのフィールド要素が示される。



Fig.3 データソースのアタッチウィザード



Fig.4 データソースの構成

データソースが構成できれば次はデータを表示する画面を構成する。デザイナーで「生徒」データソースを表示したままメニューから「画面を作成します」をクリックするか、画面右上の「ソリューションエクスプローラー」の「画面」要素を右クリックし「画面の追加」を実行する。

「新しい画面の追加」ウィザードではいくつかの画面テンプレートと、紐づける「画面データ」を選ぶことで画面を構成できる。求められる作業に応じて最もふさわしい画面テンプレートを選ぶことになるが、ここではまず生徒情報を登録するための画面を作ることとし、表形式でデータを編集できる「編集可能グリッド画面」を選択する。紐づける画面データは生徒テーブルを選択するが、このときデータ接続の名前は「testData. 生徒 Item」のような名称になっている。

「生徒」データソースとそれに紐づいた画面が構成できれば SQL Server に接続しデータを編集できるクライアントアプリケーションは完成する。デバッグを開始すると完成したアプリケーションを確認することができるが、最初のビルドではコードを自動生成するプロセスが行われるため、ビルドにやや時



Fig.5 新しい画面の追加

間がかかる。

完成したアプリケーションの画面は表計算ソフトに似ており、操作に迷うことはない。データベースからデータを読み出すには画面左上の「最新の情報に更新」アイコンをクリックし、新しいデータを登録するには表に必要な項目を入力して「保存」アイコンをクリックする。表の入力形式はフィールドのデータ型に応じて自動的にコントロールが割り当てられ、生年月日のような日付型の場合はカレンダーコントロールで入力ができるようになっている。

ビルドした LightSwitch アプリケーションで何件かの生徒情報を入力し「保存」ボタンをクリックして SQL Server にデータを送信する。これを SQL Server Management Studio で確認する。



Fig.6 デザイナーで構成された画面



Fig.7 ビルドしてデータ入力を行う



Fig.8 SQL Serverに登録されたデータの確認

7. LightSwitch の可能性

業務システム開発において、業務プロセスは現場の人間がいちばんよく知っており、開発工程ではプログラミングとデータベースの知識が必要である。システム開発には業務の理解とプログラミングやデータベースの理解の両方がなければならない。求められる機能が複雑化し、要求定義が肥大化する中でウォーターフォール型の大規模プロジェクトが行き詰まる一方で「アジャイルソフトウェア開発」のようなモデルが現れ、OBA 開発のような現場主体の開発手法が使われている。学校の校務システム、成績処理システムを構築する場合においても、このような LightSwitch と SQL Server を使ったシステム開発は素早く構築でき柔軟性が高い。SQL Server は Express 版が無料で利用できるし、LightSwitch 2011 は 90 日間利用できる無償の評価版を使うことができる。また今後もこのような「素早い開発」ができる開発ツールが徐々に世に現れ、開発環境は大きく変化すると予想される。生徒たちにも機会があればこのようなツールに触れさせ、プログラミングに興味を持たせたい。

参考文献・URL

- 1) Microsoft「Visual Studio LightSwitch」, <http://www.microsoft.com/japan/visualstudio/lightswitch>
- 2) Microsoft「Visual Studio LightSwitch 評価版のダウンロード」, <http://www.microsoft.com/japan/visualstudio/lightswitch/try>
- 3) Microsoft「SQL Server® 2012 Express のダウンロード」, <http://www.microsoft.com/ja-jp/download/details.aspx?id=29062>
(以上いずれもアクセス日は 2013 年 3 月 31 日)
- 4) アンドリュウ・プラスト『Visual Studio LightSwitch テクニカル ホワイト ペーパー』Microsoft, 2011 年