

ビジュアル開発環境BoxBasicによるプログラミング教育

山 下 明 博

Programming Education by Visual Development Environment BoxBasic

Akihiro YAMASHITA

要 旨

筆者は、ほぼ全員がプログラミングの初習者であるという科目を担当することになった。そこで、初習の学生でも、プログラミングの概念を理解し、プログラミング言語でプログラムを容易に作成できることを目標に、ビジュアル開発環境BoxBasicを開発した。この開発環境は、「ボックス」と呼ぶ長方形を組み合わせ、マウスを使用することにより、手軽にプログラミングを開発することができる。また、BoxBasicは、ステップごとの動作確認、C言語ソースへの自動翻訳といった機能を有し、特定のプログラミング言語に依存しないプログラミング教育を行うことができる。本論文は、ビジュアル開発環境BoxBasicの設計思想、技術的特徴と、プログラミング教育で予測される学習効果を論じたものである。

キーワード：BoxBasic・プログラミング・ビジュアル開発環境・プログラミング教育

は じ め に

コンピュータを動作させるために必要なソフトウェアを開発する人材の育成には、時間と労力が必要である。筆者は、2013年、プログラミング言語における完全なビジュアル化を実現することを目標に、BBB (Box Based BASIC) というプログラミング言語を設計・開発した。BBBは、画面上の部品を設計する部分をビジュアル化するだけでなく、これまではテキストエディタで記述していたロジックの部分のプログラミングも含めた全ての操作を、「ボックス」と呼ぶ長方形に基づいてビジュアル化することを可能にしたプログラミング言語である¹⁾。これにより、ソフトウェア開発の労力を大きく減らすことができた。

今回、安田女子大学現代ビジネス学部現代ビジネス学科において、1年生対象に、「ビジネス情報処理」という科目を担当することになった。この科目は、ビジネスに必要な情報処理技術の基礎として、プログラミング技術を教育する内容である。

しかし、ほぼ全員がプログラミングは初習であり、通常の方法では、学生の理解度が高まらない恐れがあった。

そこで、プログラミングを始めて学ぶ学生でも、プログラミングの概念を理解し、プログラミング言語でプログラムを作成できることを目標に、ビジュアル開発環境BoxBasicを開発した。これは、プログラミング言語BBB (Box Based BASIC) と共通の開発理念により開発したもの

で、「ボックス」と呼ぶ長方形をマウスで作成し、それらを組み合わせることにより、プログラミングを視覚的に開発できる環境を提供するものである。また、ステップごとに実行しながら、動作結果を知ることができたり、C言語のソースを自動生成し比較することもできる。このような機能により、特定のプログラミング言語に依存しないプログラミング技術を習得することができる。

本論文は、筆者が設計し、開発したビジュアル開発環境BoxBasicについて、その技術的特徴と、BoxBasicを使ったプログラミング教育を実施する際に使用するサンプルについて、具体例を示し、ビジュアル開発環境によるプログラミング技術の有用性を明らかにするものである。

I. 開発の容易なプログラミング環境

1. ソフトウェア生産性・保守性の向上

コンピュータを動作させるためには、ハードウェアを制御するソフトウェアをプログラミングする必要がある。そして、複雑なプログラミング作業をなるべく容易にするために、様々な言語や手法が開発されてきた。機械語、アセンブラ言語、FORTRAN、COBOL、BASIC、PASCAL、C、C++、Javaといったプログラミング言語の開発、構造化プログラミング、オブジェクト指向プログラミングといった生産性を高める手法の開発が行われ、大規模なソフトウェアの開発に適用することにより、ソフトウェアの生産性・保守性が向上した。

2. ビジュアルプログラミング言語

視覚的な操作によりプログラミングを行うという考え方は、ソフトウェアの生産性向上に非常に高い効果を有する。その代表が、Visual BASIC言語²⁾である。1995年のWindows95発売以降、コンピュータにおいては、GUI (Graphical User Interface) が主流となり、Windows上で動作するプログラミングを開発しようとするプログラマが急増した。Windows上でプログラミングを行うには、当時、C++というテキストプログラミング言語を使用し、非常に難解なMFC (Microsoft Foundation Class) というクラスライブラリ³⁾を理解して使用する必要があった。しかし、それまでMS-DOSの世界でプログラミングをしていたプログラマにとって、MFCは、全く新しい概念であり、しかも難解なため、プログラミングを諦めるユーザも多く存在した。

そのような状況下において、次第に注目を浴び、急速にユーザを増やしていったのが、Visual BASICというプログラミング言語である。Visual BASICは、古くから存在するBASIC言語との親和性を有し、マウスを使ってWindows部品を容易に作成できる手軽さを有しているため、初心ユーザでもすぐにプログラミングすることが可能であり、多くのユーザに受け入れられた。プログラミングは、ツールボックスから、使用したい部品の種類を選び、マウスで画面上をドラッグするだけであり、ラベル、テキストボックス、チェックボックスといった様々なWindows部品を容易に作成することができた。

しかし、Visual BASICにもビジュアル化されていない部分が存在する。Visual BASICは、マウスを使ってビジュアルにWindows部品を作成できるが、Windows部品に入力があった場合に行うべきイベント駆動の部分の動作は、テキストエディタでプログラミングしなければならない。その意味で、Visual BASICは不完全な半ビジュアル言語であるといえる。

II. 完全なビジュアル開発環境BoxBasic

1. 完全なビジュアル言語

筆者は、完全なビジュアルプログラミング開発言語として、BBBを開発した。完全なビジュアル言語の定義は、画面上の部品を作成する部分をビジュアル化するだけでなく、ロジックの部分のプログラミングもビジュアル化した言語のことである。

その実現のために採用したのが、全ての操作を、画面上に「ボックス」と呼ぶ長方形を作成し、これによりロジックの記述を行う手法である。

今回開発した、BoxBasic開発環境は、BBBの考え方を発展させ、完全なビジュアル言語でプログラミングを行い、さらに、プログラム動作確認、デバッグ、他のプログラミング言語への翻訳まですべてが一体的に行える環境を用意するものである。

2. ロジックプログラミングに必要な要素

BoxBasicでは、14個のロジックプログラミングに必要な基本要素を組み合わせ、ビジュアルにロジックの部分プログラミングすることが可能とした。

- A) 処理：処理のまとめ
- B) 変数：動作中に値を保持する領域（数値型と文字型）
- C) 配列：動作中に添え字により複数の値を保持する領域（数値型）
- D) 計算式：変数に操作を加える式
- E) 条件式：反復や分岐を行うときの真偽を評価するための式
- F) 引数：処理を呼出するときに引き渡すデータ群
- G) 入力：外部からのデータ入力
- H) 出力：外部へデータ出力
- I) 反復：条件式に従って真偽を評価し、反復して計算式を実行する部分
- J) 分岐：条件式に従って真偽を評価し、計算式を実行する部分
- K) 脱出：反復から脱出する文
- L) 終了：処理を終了する文
- M) 呼出：他の処理を呼び出す文
- N) 復帰：処理を終え、呼出元に制御を戻す文

BoxBasicは、プログラミング未習の初心者を対象に教育を行うことを目的としている。そこで、必要最小限の機能を盛り込むようにした。例えば、配列で文字型を扱った場合、C言語では、2次元配列となり、初心者には難し過ぎると思われる。また、配列で文字型を扱う処理を行う頻度も非常に少ない。そこで、BoxBasicでは、変数は数値型も文字型も扱うが、配列では数値型のみ扱うように限定した。

3. 特徴的な機能

(1) 他のプログラミング言語ソース出力機能

BoxBasicでは、ネイティブモードでの実行のほかに、他のプログラミング言語のソースを出

力する機能を持たせている。それは、BoxBasicでプログラミングを行うとき、他の言語と比較したほうが、プログラミング技術の理解が進むからである。現在、BoxBasicは「C言語」への翻訳をサポートしている。これにより、自分が作成したBoxBasicのプログラムをC言語で記述するときはどのように記述すればよいかが一目でわかる。また、C言語は、Windowsパソコンだけでなく、UNIXワークステーションや、FAパソコンといった、多くの環境の下で動作するため、BoxBasicでプログラミングしたプログラムを、多くの環境下で動作させることも可能であり、高い移植性が確保できる。

(2) 変数管理の自動化

プログラミング言語において、変数名の管理は重要である。変数を宣言せずに使用できる言語も存在するが、多くの言語においては、変数を宣言した後に使用する必要があり、プログラマにとって、そのことが大きな負担である。

BoxBasicでは、そのような負担を軽減するために、変数管理を自動化している。例えば、プログラマが計算式や条件式の中で変数を使用した場合、同名の変数宣言を行っていないければ、変数を宣言するボックスを自動的に生成する機能を持っている。これにより、変数名の重複や宣言忘れを防止することができる。

このように、変数管理を自動化することにより、プログラマの負担を軽減できることが、BoxBasicの大きな特徴である。

Ⅲ. BoxBasicの画面と機能

1. BoxBasicの画面

BoxBasicを起動すると、図1のような初期画面が表示される。

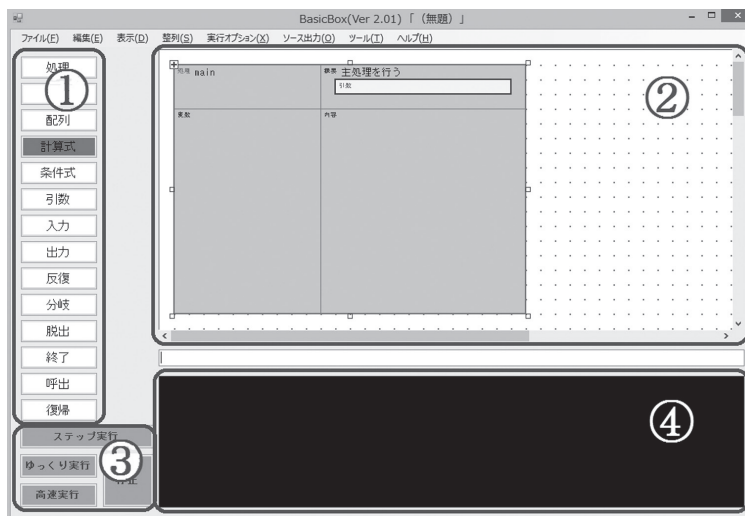


図1 BoxBasicの初期画面と4つの領域

その中に、①～④の4つの領域があり、これらが協働して、ビジュアルプログラミングを行う。

①～④の領域が果たす役割は、以下のとおりである。

①ロジックプログラミングのための基本要素

14種類のボタンは、ロジックをプログラミングするためのものであり、いずれか一つをクリックすると、ボタンの背景が赤くなる。その状態で、②のボックス作成域上でマウスをドラッグすると、①で選んだ種類のボックスが作成される。

②ボックス作成域

この上で、複数のボックスを作成し、それがプログラムのロジックを表す。ボックスは、左上にあるものから、右下にあるものへと順番が付けられ、その順に実行される。実行時には、実行しているボックスの色がピンクに変わるので、どこを実行しているかが一目でわかるようになっている。

③実行ボタン

②のボックス作成域で記述したプログラムを実行するためのボタンを用意している。

「ステップ実行」：1つのボックスずつ、ステップを追ってプログラムを実行し、④実行結果表示域に、ステップ表示と共に実行結果を表示する。

「ゆっくり実行」：1秒のインターバルで、連続してプログラムを実行し、④実行結果表示域に、ステップ表示と共に実行結果を表示する。

「高速実行」：インターバルなしで、連続してプログラムを実行し、④実行結果表示域に結果のみ表示する。

「停止」：「ゆっくり実行」および「高速実行」中に、プログラムを停止させる。

④実行結果表示域

③実行ボタンにより実行した結果を表示する領域である。「高速実行」の場合は、「出力」ボックスの結果などを表示するが、「ステップ実行」「ゆっくり実行」の場合は、結果に加え、ステップ表示も行う。

Ⅲ. BoxBasicを用いたプログラミング教育方法

BoxBasicを用いプログラミング教育は、11種類のサンプルプログラムにより行う。11種類のサンプルプログラムは、プログラミング技術に最低限必要な要素を含んでいるため、これらを順序よく動作確認しながら学習を進めることにより、プログラミング技術を非常に短時間で身に付けることができる。

BoxBasicを使い、学ぶことができるのは、プログラミング技術の中で、「処理」「変数」「演算子」「文字列」「配列」という基本要素、「順次処理」「分岐」「反復」という基本制御、「脱出」「終了」「呼出」「復帰」という基本制御からの逸脱、「計算式」「条件式」という式、「入力」「出力」という入出力についてである。

表1に、サンプルプログラムと、それにより学ぶことができる主要素を示す。

表1 サンプルプログラムと学ぶことができる主な要素

サンプルプログラム名	学ぶことができる主な要素
SAMPLE01_HelloWorld出力.LAY	処理, 出力, 文字列
SAMPLE02_変数.LAY	(数値型) 変数, (代入) 演算子, (算術) 演算子, 計算式
SAMPLE03_順次処理.LAY	順次処理
SAMPLE04_分岐と条件式.LAY	分岐, 条件式, (比較) 演算子
SAMPLE05_反復.LAY	反復
SAMPLE06_脱出.LAY	脱出
SAMPLE07_終了.LAY	終了
SAMPLE08_入力.LAY	入力
SAMPLE09_文字型変数.LAY	(文字型) 変数
SAMPLE10_配列.LAY	配列
SAMPLE11_呼出と復帰.LAY	呼出, 復帰

以下, 11種類のサンプルプログラムのうち, 代表的なものを抜粋して説明を行う。

1. SAMPLE01_HelloWorld出力.LAY

SAMPLE01_HelloWorld出力.LAY は, プログラミングで最も初めに学ぶといわれている「Hello World」という文字列を出力するプログラムである。これまで, 多くのプログラミング言語上で, 「Hello World」を出力することが行われてきた。

「SAMPLE01_HelloWorld出力.LAY」プログラムは, BoxBasicにおいて, 「Hello World」を出力するためのプログラムである。そして, このプログラムを通じて, 「処理」ボックスの使い方, 「出力」ボックスの作り方, ダブルクォテーションで囲んだ「文字列」の使い方を学ぶことができる。

以下に, BoxBasicの場合に, 「Hello World」を出力するまでのプログラム手順を述べる。

(1) 初期画面での「出力」ボタン選択とボックスの作成

BoxBaicの初期画面で, 左に並んだボタンの中から「出力」をクリックすると, 「出力」ボタンの背景が赤くなる。その状態で, 「処理」ボックスの中の右下の「内容」サブエリアに, マウスの左ボタンでドラッグすることにより, 「出力」ボックスを作成する。図2に, 初期画面での「出力」ボタン選択と「出力」ボックスの作成の様子を示す。

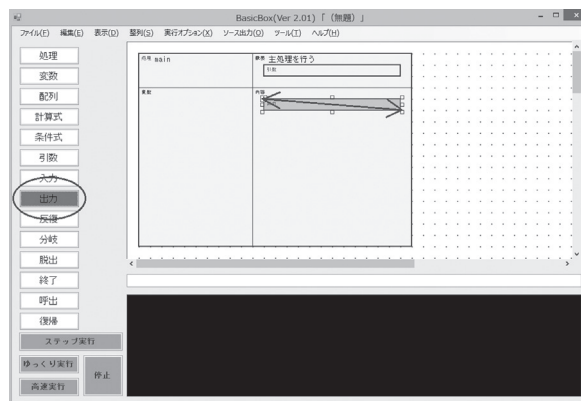


図2 初期画面での「出力」ボタン選択と「出力」ボックスの作成の様子

(2) 「出力」の「文字列」部分を入力する画面への "Hello World!" の入力

BoxBasicで、「出力」ボックスをマウスの左ボタンでダブルクリックすると、図3のように、出力の「文字列」部分を入力する画面が開く。そこで、「Hello World!」と入力し、決定をクリックする。

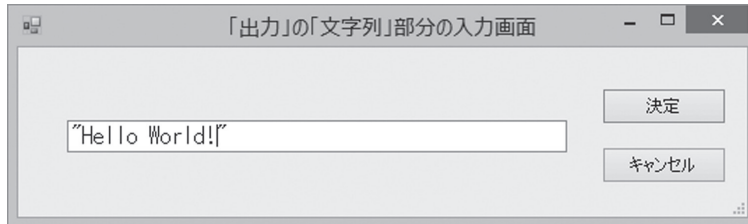


図3 出力の「文字列」部分を入力する画面

(3) 「HelloWorld出力」プログラム作成結果

以上の操作により、「HelloWorld出力」プログラムが完成した。

図4に、「HelloWorld出力」プログラムを示す。

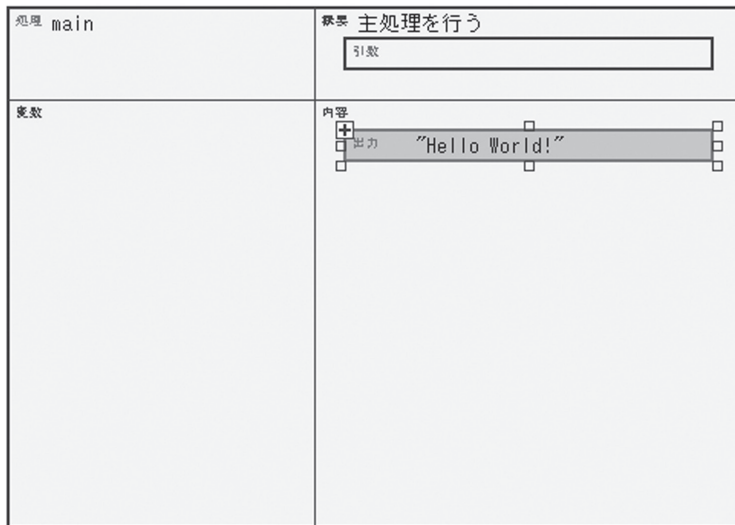


図4 「HelloWorld出力」プログラム

BoxBasicでは、このように、ボックスの組合せだけでプログラミングを行う。そのため、プログラミングが簡単に行える。

(4) 「HelloWorld出力」プログラムの高速実行

BoxBasicで、「HelloWorld出力」プログラムを実行するには、左下の「高速実行」ボタンをクリックする。

その結果、図5のように、「Hello World!」という実行結果が表示される。



図5 「HelloWorld出力」プログラムの実行結果

(5) C言語への変換

BoxBasicの「ソース出力」「Cソース出力」メニューを選択すると、「HelloWorld出力」プログラムは、C言語に翻訳され、その結果が表示される。

図6に、C言語に翻訳された「HelloWorld出力」プログラムのソースを示す。

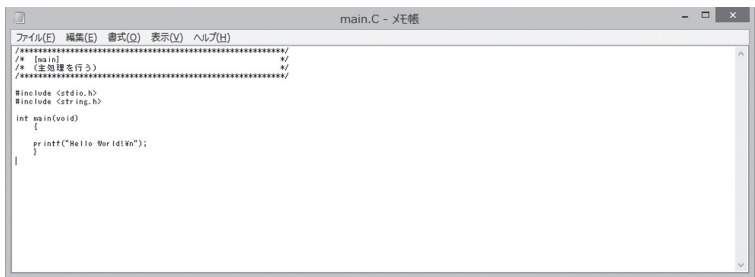


図6 C言語に翻訳された「HelloWorld出力」プログラムのソース

2. SAMPLE09_文字型変数.LAY

「SAMPLE09_文字型変数.LAY」プログラムは、BoxBasicにおいて、「文字型変数」処理を学ぶためのプログラムである。

(1) 文字型変数プログラム

変数Sと変数Tに文字列を入力し、それを組み合わせて出力するプログラムである。このプログラムの高速実行の結果は、BoxBasicの場合、図7のようになる。

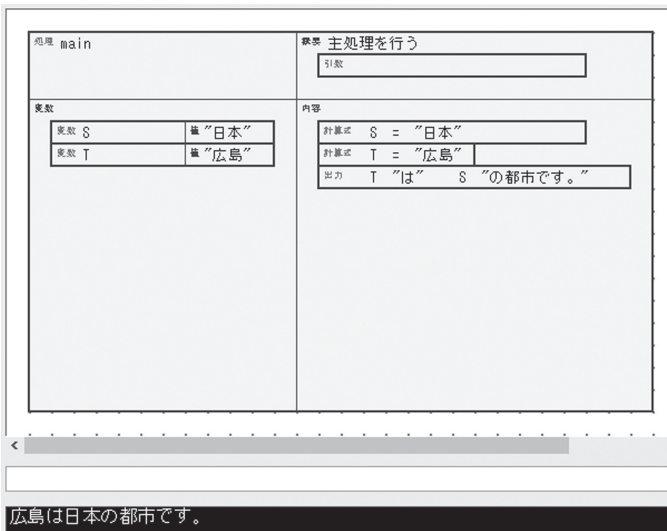


図7 文字型変数プログラムと高速実行の結果

(2) C言語への変換

BoxBasicの文字列変数プログラムをC言語に翻訳した結果を、図8に示す。

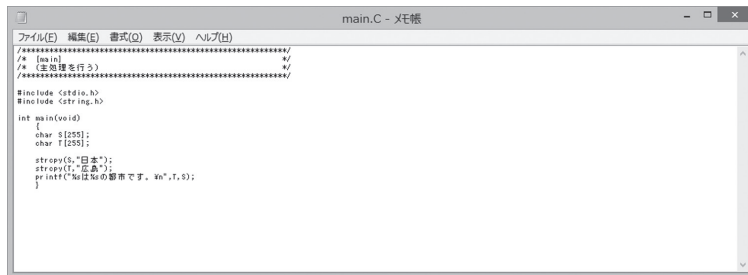


図8 C言語に翻訳された文字列変数プログラムのソース

結 論

BoxBasicは、初めてプログラミングを学ぶ学生でも、容易にプログラミングを開発できる環境を提供するものである。BoxBasicは、これまではテキストエディタで記述していたロジックの部分のプログラミングを、画面上に作成する「ボックス」と呼ぶ長方形によってビジュアル化した。また、BoxBasicで使用する基本要素を14個に限定し、それらの要素を組み合わせでプログラミングすることにした。これにより、初習の学生でも混乱せず、確実にプログラミングの技術を身に付けることができると考える。

また、サンプルを11個用意した。学生は、これらを一通り作成し、動作させてみることで、順序を追って、プログラミング技術を学ぶことができる。そして、BoxBasicでプログラミングした結果をC言語に翻訳し、両者を比較し、共通点を理解することができるため、特定の言語にとらわれない汎用的なプログラミング能力を身に付けることができると考える。

実際には、「ビジネス情報処理」で、学生がプログラミングを行うのは、平成26年9月以降であり、BoxBasicを使って学生に指導する上で、問題点が生じることも予想される。学生の反応を得ながら、BoxBasicの機能をさらに拡充し、多くのプログラマに使ってもらえるようなプログラミング開発環境に成長させていきたいと考える。

注

- 1) 山下明博 (2014), 「完全なビジュアルプログラミング言語BBBの設計と開発」, 2013年度安田女子大学紀要第42号, p.231 ~ p.247。
- 2) Visual BASICは、Microsoft社が販売したプログラミング開発環境であり、当初はMS-DOS環境下でのビジュアルなプログラミングツールとして開発された。その後、1995年にWindows95が発売されると、Windowsアプリケーションソフトウェアを手軽に開発できるプログラミング言語として注目され、多くのWindowsアプリケーションがVisual BASICによって開発された。特に、1998年に発売されたVisual BASIC 6.0は、データベースアクセス機能、ActiveX作成機能など、高度なプログラミングも可能となり、高い評価を得た。
- 3) 吉田弘一郎 (1994), 「Visual C++によるMFCライブラリの使い方」参照。

[2014. 9. 25 受理]