

研究ノート

Visual Basic .NET でのプログラミング教材作成

Programming material making with
Visual Basic .NET堀 合 公 威*
HORAI, Kouji

I. はじめに

大学における高度に進んだ教育は、整備されたマルチメディア環境とその資源を高度に活用した、充実した内容の授業がダイナミックに展開されるものとする。それを実現するためには、マルチメディア教室およびアクセスポイントの充実と、その授業展開を下支えする教材作成支援システムが、充実していることである。

これまで本学では、上記のシステムを目指して、情報科学研究センターを中心に、平成 16 年度に高速な無線・有線 LAN 環境の整備、平成 17 年度に情報教育教室の整備を行ってきた。今年度整備された情報教室を利用した情報教育関連の講義は、それぞれの講義担当教員の努力で環境に適応した講義の充実がはかられている。一方通常の講義においては、マルチメディア教室を手軽に利用して、タイムリーにマルチメディア教材を提示する充実した授業を行うことが求められており、そのマルチメディア教材の作成および蓄積が急がれる。

いつでも・どこでも・何とでも（誰とでも）に代表される高度に進んだユビキタス社会では、ウェアラブルコンピュータが活用されると言われており、教育においても来るべきウェアラブルコンピュータ時代への対応が次の問題である。そしてその活用を意識したマルチメディア教材の作成が問題となるであろう。

しかし、現在はその前段階であると捉えており、その前触れである軽量なモバイルコンピュータは、まだまだ高価であり一般に普及していない。モバイルコンピュータを活用した、魅力あるマルチメディア教材が提供されて、それを持つことによるメリットが十分認識されるのであれば、多少高くても学生は個人購入し活用が進むと考えるが、いまだ学内に魅力あるマルチメディア教材の蓄積が十分でなく、学生が軽量なモバイルコンピュータを持つ必然性が感じられないのが実情である。

* 城西大学理学部化学科

したがって今年度の新システム導入では、そのような提案（学生の個人購入の促進）を積極的に出来なかったことが最大の反省点だと思っており、マルチメディア教材の蓄積およびデータベース化の促進と共に教材利用環境の整備が重点的に行われた。学内には十分とは言えないが、教材作成のツールが存在しそれらを活用すれば、工夫しだいで充実したマルチメディア教材を作成することは可能であると考えられる。作成されたマルチメディア教材を活用するためには、教材を配信する環境（サーバーなど）が必要でありその整備に重点がおかれた。また、現在ある教育用コンテンツを手軽に活用し学内外の教材資源（高価なソフト・データベース・インターネットの利用・電子教材など）を利用できるような環境整備が行われると共に、それをサポートする柔軟の思考を持った若いスタッフが必要であると考え、それを解決する一つ的手段として、このシステムの管理およびこのシステムを使用して教育する教員への e-Learning アドバイザーとして、導入業者から常駐の技術者を派遣されており、城西大学の教育全般に、このシステムの機能が十二分に活用される期待される。

今回はこのような背景を元に、下記の既存の教材のマルチメディア教材化への試みを紹介する。

大学向け授業支援システム Campusmate/CourseNavig システムの活用したマルチメディア教材 e-Learning コンテンツの作成

1. 情報教育：情報科学 B Visual Basic 6.0 教材から Visual Basic .NET 教材へ
プログラミング教育環境は、旧システムの Microsoft Visual Studio 6.0 から、新システムの Microsoft Visual Studio .NET 2003 になったための大幅な教材の修正および CourseNavig システムへの登録
2. その他の教材：既存教材のマルチメディア教材化および CourseNavig システムへの登録

II. プログラミング教育

1. 情報科学 B における Visual Basic .NET でのプログラミングについて

情報教育におけるプログラミング教育は、様々なプログラム言語で行われており、情報分野の資格試験に対応するには JAVA や C 言語を用いて行うのが良いと考えている。しかし、プログラミング教育を本格的に受けていない化学科の学生にとって、最初からそのような言語でのプログラミング教育は、ハードルが高いのではないかと考えている。

Visual Basic .NET は、元は Basic 言語から発展した言語なので、簡単なソフトの起動法と文法および画面への描画コマンドを教えることで、受講している学生は、プログラムの作成作業に移行できるのが良い点であると考えられる。

図 1 は、その説明で用いたパワーポイントの画面の一部であり、また、プログラムの作製課題に

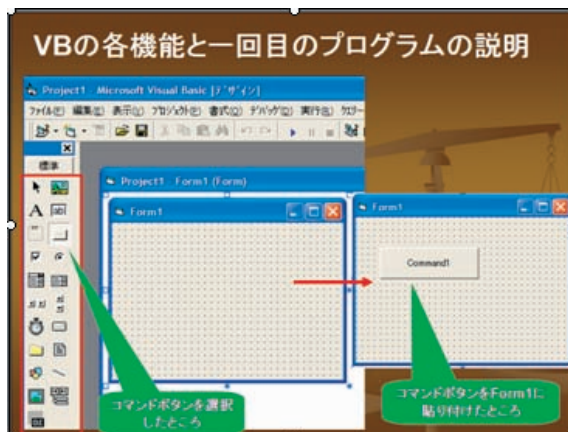


図1 ソフトの起動法とプログラミングの説明画面

もよるが、Visual 的なプログラム作製課題を与えるならば、プログラムを作成・実行・実行結果の確認までの作業を、視覚で確認できることが、学生には理解されやすいと考えている。

2. Visual Basic 6.0 教材について

これまで情報科学 B の講義は、清光会館 3-302 演習室において Microsoft が販売している統合プログラム作成ソフト Microsoft Visual Studio 6.0 のパッケージソフト中の Visual Basic 6.0 を用いて Visual 的なプログラミング教育を行ってきた。しかし、平成 17 年夏に行われた新システムの移行の際に Microsoft Visual Studio .NET 2003 のパッケージソフトなり、それに伴って Visual Basic 6.0 がネットワークを意識したプログラム作成ソフト Visual Basic .NET へアップグレードした。それにより、これまでの教材用課題ソフトに生じた不具合に対応するものである。

このソフトの変更に伴い、Visual Basic .Net は .NET Framework を基盤として、種々のアプリケーションや XML Web サービスコンテンツの作成・実行ができるようになった。しかし、従来の Visual Basic 6.0 で作成したプログラムは、Visual Basic .NET ではそのままでは実行できなくなりアップグレード作業が必要であった。もっとも大きな変更点は PictureBox など図を描く時のコードの変更、ツールの名前の変更・追加などであり、アップグレードでは対応できないため、全面的に作成し直す必要があった。

例として Visual Basic 6.0 で作成したモンテカルロ法のフォーム画面（図 2）とプログラムソースを紹介する。図 2 では、フォーム上に計算実行ボタンと作図領域用ピクチャーボックスが貼り付けてある。

図 3 は、その実行後の画面である。約一万点の点が、1/4 円内では赤、それ以外は黄色で表示されている。図 2 中の計算実行ボタン（コマンドボタン）に、下記のようなプログラムコードが入力されている。

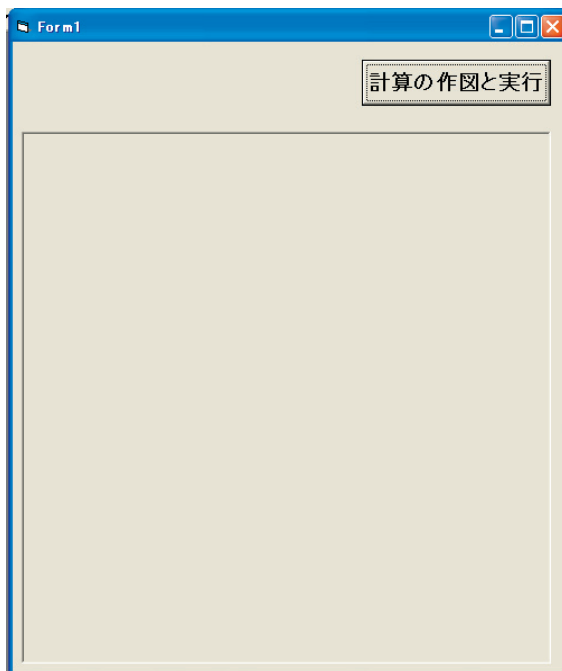


図2 Basic 6.0 で作成したモンテカルロ法のプログラムのフォーム画面

Visual Basic 6.0 で作成したモンテカルロ法のプログラムコード

```
Private Sub Command1_Click()
Visible = True

Dim a As Long, m As Long, c(0 To 1) As Double, p As Double
Dim I As Double, x(0 To 1) As Double, y(0 To 1) As Double
Dim d(0 To 1) As Integer, e(0 To 1) As Integer
Print "パイの計算と作図"
Print "モンテカルロ法による  $\pi$  の計算とグラフ"

red = QBColor(12)
XXX = QBColor(6)
Picture1.ScaleWidth = 1
Picture1.ScaleHeight = 1
Picture1.Circle(0, 0), 1
For I = 0 To 1 Step 0.0001
y(I) = Rnd()
x(I) = Rnd()
c(I) = x(I)^2 + y(I)^2
```

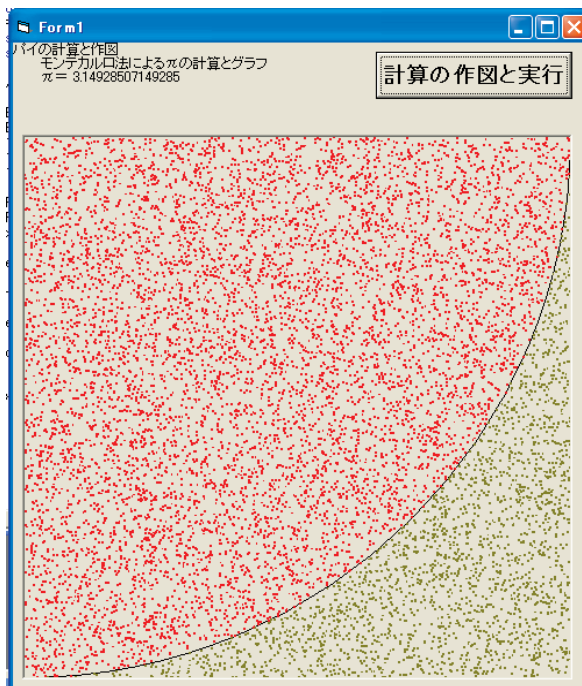


図3 Basic 6.0 で作成したモンテカルロ法のプログラムの計算実行後の画面

```

If c(I) <= 1 Then
    Picture1.Circle(x(I), y(I), 0.001, red)
    d(I) = c(I)/c(I)
    a = a + d(I)
Else
    Picture1.Circle (x(I), y(I)), 0.001, XXX
End If
e(I) = c(I)/c(I)
m = m + e(I)
Next I
p = (4 * a)/m
Print "π = "; p
End Sub

```

このように、Visual Basic 6.0 で作成したプログラムコードは、シンプルで、プログラミングの入門教育では最適であった。また、様々な計算用プログラムや卒業研究用プログラムの作成に利用されてきた。しかし先の事情で、Visual Basic .NET にプログラミングの入門教育を移行する必要

が生じた。

3. Visual Basic 6.0 教材から Visual Basic .NET 教材へ

先で紹介した Visual Basic 6.0 で作成したプログラムは、Visual Basic .NET 実行できないため、新たに作成したが、プログラムは、かなり複雑になった。図 4 は、Visual Basic .NET で作成したモンテカルロ法による π を求めるプログラムの実行画面であり、フォーム中に計算開始ボタンとピクチャーボックスが配置されており、これは Visual Basic 6.0 の場合と同じである。しかし、フォーム中直接表示する Print コマンドが無くなりそれに対応するためラベルとテキストボックスを配置してある。その他にフォーム上に配置したオブジェクトには、プロパティを厳密に定めなければならないなど、面倒な点が増えている。

次に計算開始ボタンに入力したプログラムコードを示す。

Visual Basic .NET で作成したモンテカルロ法のプログラムコード

```
Private Sub GoButton_Click(ByVal sender As System.Object, ByVal e As System.  
EventArgs)
```

前の行の続き

Handles GoButton.Click

```
DisplayEllipse(LinePictureBox)
```

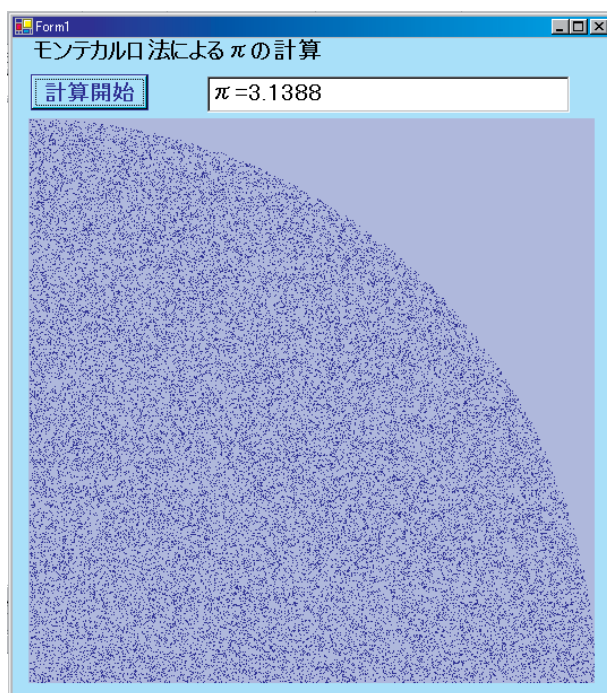


図 4 Visual Basic .NET で作製したモンテカルロ法の実行画面

```

End Sub
Sub DisplayEllipse(ByVal p As PictureBox)
    Dim i As Object
    Dim XXX As Object
    Dim red As Object
    Dim pai As Object
Dim g As Graphics = p.CreateGraphics()
    Dim PinkBrush As SolidBrush = New SolidBrush(Color.Pink)
    Dim BlueBrush As SolidBrush = New SolidBrush(Color.Blue)
    Visible = True
    Const N As Integer = 100000
    Dim b, a, c As Double
    Dim X, y As Single
    For i = 1 To N
        X = Rnd()
        y = Rnd()
        If X * X + y * y <= 1 Then
            c = c + 1
        End If
    Next i
    pai = c * 4/N
    Text1.Text = "π=" & pai
End Sub
Private Sub text1_TextChanged(ByVal eventSender As System.Object, ByVal event
Args
    前の行の続き As System.EventArgs) Handles Text1.TextChanged
End Sub
End Class

```

Visual Basic 6.0 で作成したプログラムでは、コマンドボタンの実行コマンドでは

```
Private Sub Command1_Click()
```

であったが Visual Basic .NET では

```
Private Sub GoButton_Click(ByVal
    sender As System.Object, ByVal
    e As System.EventArgs)
    Handles GoButton.Click
```

となり複雑になっており、色の指定も Dim red As Object などこのように定義する必要があるなど、多くの修正が必要であった。

4. Visual Basic .NET 教材の活用

3. で紹介したモンテカルロ法のプログラムは、乱数関数を使って X 軸および Y 軸の値を求め、その点が $1/4$ 円内に入るかを判定し n 回中何回入ったかをカウントし π を求めるものである。図 4 では、円内に入った点を濃い青い点で表示した。ここに示した実行画面では、1 万回繰り返しから求めた π の値は 3.1388 で精度の悪い値であった。それに対して、図 5～図 7 に積分法で π の値を求めるプログラムに実行画面を示した。

図 5 は、 x 軸（円の半径）を 1 万分割して計算した場合で、繰り返し計算の回数は図 4 の場合と同じである。これらの図は、左下を原点 $(0, 0)$ とした場合のそれぞれの x 軸の点における $1/4$ 円弧上の値を求め、図 7 で示すように黒線で囲ったそれぞれの長方形の面積を求め合計が $1/4$ 円の面積と仮定して π の値を求めるものである。

図 7 は、たった 5 分割で求めた π の値では、3.172 で、ある程度の値が求められることが分かるが、実際に精度が悪いことが実感できる（円弧上の値を、 x 軸の点 $h/2, 3h/2, 5h/2, \dots$ と h 分ずつ変化させて 5 回繰り返して求めている）。

図 6 は、半径を $1/15$ 等分（15 回の繰り返し計算）したときの図で、モンテカルロ法で求めた 1 万回の繰り返し計算で求めた π の値と同程度の精度で π の値を求めることができた。

繰り返し計算の回数を徐々に多くしてプログラムを実行することで、求めた π の値の精度の違いを簡単に比べることができる。また、モンテカルロ法の場合は、点が増えることで徐々に $1/4$ 円内が塗りつぶされていくことが実感でき、積分法の場合は、繰り返し計算の回数（分割数）を少ない値から徐々に大きな値に変更することで、階段状の図が徐々に円弧に近づきそれと共に π の精度が上がるのが実感できる。

計算の実行例として、半径の一万分割では 7 桁の精度で π の値が求められる。

正解

$$\pi = 3.141592653589793238462643$$

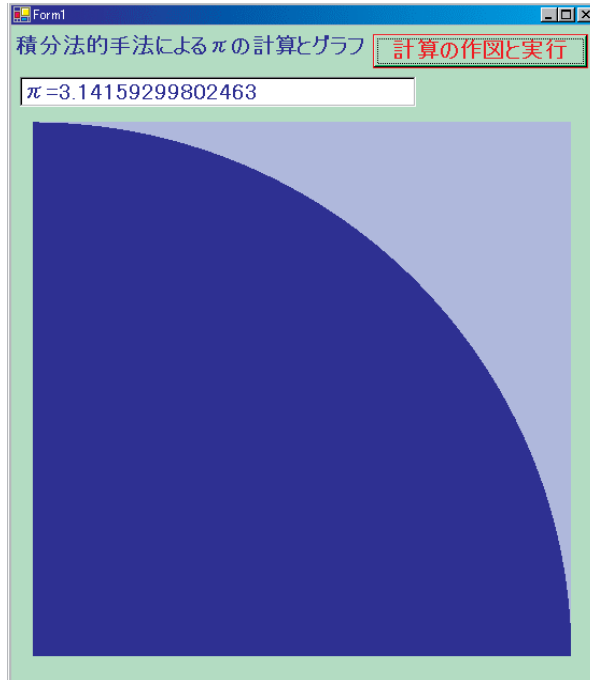


図 5 Visual Basic .NET で作成した積分法の実行画面 (1 万分割)

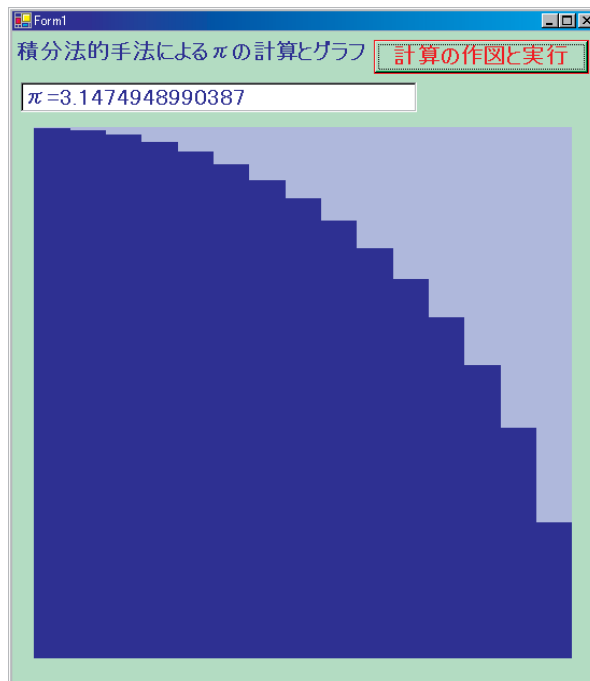


図 6 Visual Basic .NET で作成した積分法の実行画面 (15 分割)

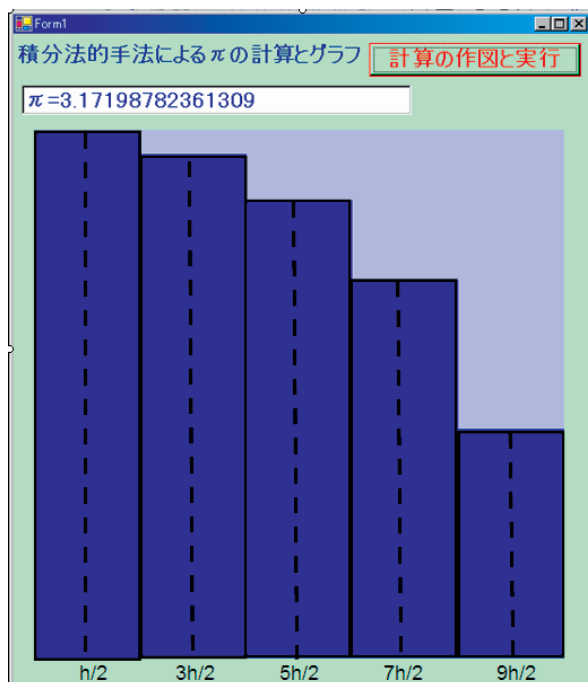


図 7 Visual Basic .NET で作成した積分法の実行画面 (5 分割)

1 万分割で計算した時の解 $\pi=3.14159299802463$

また、繰り返し計算の回数を同じにしてプログラムを実行した場合、計算方法の違いによって求める π の値の精度を簡単に比較することができる。従って、このプログラムを完全に理解すれば、プログラミングの知識およびコンピュータの利点が理解でき興味が沸くと思っている。

Visual Basic 6.0 から、Visual Basic .NET へのプログラムの変更は、先に述べたようにコマンドその他に様々な変更があり、Visual Basic 6.0 のプログラミングに慣れた者には、かなり手間が掛かる作業であった。また、ここで紹介した以外のプログラミング課題も、Visual Basic 6.0 から、Visual Basic .NET に、変更が完了している。

Ⅲ. プログラミング教育の今後

Ⅱ. で述べたように、プログラミング教育は、情報分野の資格試験に対応するには JAVA や C 言語を用いて行うのが良いと考えているが、プログラミングをする前に、最低限の言語教育が必要であり、化学科の学生にとって、必ずしも本格的なプログラミング教育は必要ではなく、プログラミングにより緻密な思考能力が養われることが第一の目標だと思っている。

Visual Basic .NET は、元は Basic 言語から発展した言語なので、簡単なソフトの起動法と文法

および画面への描画コマンドを教えることで、受講している学生がすぐに、プログラムの作成作業に移行できるのが良い点であると考え。

化学科で学ぶ学生の進路は様々であり、情報関連の企業に進む学生や一般企業の中でも会社の情報システムを管理する部門仕事に携わることが多い現状を考えるならば、当初の目標が達せられた後は、将来就職を考えている学生に対する本格的なプログラミング教育およびネットワークその他のシステム管理など情報資格試験につながるカリキュラムが必要である。

プログラミング教育で言えば、JAVA や C 言語の教材を開発する必要があると考えており、そのような言語による教材を徐々に作成し蓄積したいと思っている。また機会があれば後日紹介したいと思っている。

IV. マルチメディア教材 e-Learning コンテンツへの作りこみ

e-Learning 教育を推進するため、今年度新たに大学向け授業支援システム Campusmate/ CourseNavig が導入された。このシステムを活用して魅力的な授業を展開し、授業に於ける教育効果を高めたいと考えている。今年度は、授業に活用することには間に合わなかったが、来年度からは、このシステムを講義に本格的に活用し、授業に於ける教育効果を高めたいと考えている。

したがって、現在これらのプログラミング教材の登録作業を進め、マルチメディア教材としての活用を目指している。この教材の登録作業を手始めに、他の講義の教材も順次マルチメディア教材化して e-Learning コンテンツの蓄積を行い授業での活用を目指したい。

V. 今後の課題

今回は、「Visual Basic .NET でのプログラミング教材作成」ということで、現在の私の取り組みを紹介した。導入された CourseNavig を利用した、マルチメディア教育の実を上げることが求められており使命だと感じている。しかし、マルチメディア教材は、使用するタイミングを誤ると逆効果であることがしばしば遭遇しており、効果的に利用することがいかに難しいかをつくづく感じている。教員が自ら作成した教材に、伝えたい内容が凝縮されていれば、マルチメディア教材として提示することは、たとえ稚拙な技術で作成した教材であっても、エレガントに作成された市販のマルチメディア教材だけを利用することよりも、作り手の熱意が伝わりそのことが非常に大事だと思っている。したがって、これからは機会をみて先生方に、マルチメディア教材の活用例を紹介し CourseNavig ユーザーを増やしたいと思っており、この紹介例を参考にいただければと思っている。

そしてマルチメディア教材活用の輪が広がることを期待している。また、今後自らマルチメディ

ア教材を作成してみたいと考えているが、具体的な手続きなどでなかなか踏み切れない先生がおられたらご一報いただければと思っている。

まだまだ実践での試みは浅く、効果の評価はこれからであるが、マルチメディア教材の活用は、必ず効果が上がると信じて進めていきたいと思っている。

(Received Mar. 2, 2006)