原著論文

Visual Basic を用いた「GUI のデザイン&ユーザビリティ」の継承型学習

野村 典子, 太田 健一

武庫川女子大学 生活環境学部 情報メディア学科

INHERITANCE LEARNING FOR “GUI DESIGN & USABILITY” UTILIZE VISUAL BASIC

Noriko NOMURA and Kenichi OHTA

Mukogawa Women's University, 6-46 Ikebiraki, Nishinomiya, Hyogo 663-8558, Japan

Abstract: The objective of this paper is to describe a design method for instruction in computer programming. The most effective method is developing some real software to solve the familiar problem. Students can learn a lot of knowledge and skills of program comprehension and system design, developing real software as well as their own organization of the concepts at a time. We designed and practiced this method, utilizing Visual Basic programming languages making use of its easy designing for graphical user interface. Also the merits of visually program running, we can understand comparatively easy the code of program written by others. So we make inheritance learning possible with software and detailed documents archived. We have separately reported the effectiveness of this method from the reaction of the students in this class and many software output. In this paper we report the same effectiveness learning for “GUI Design & Usability”.

Keywords: Programming Education, Visual Basic, Usability

1. は じ め に

2005 年に経済産業省等が発行した『平成 15 年度ものづくり白書（製造基盤白書）』によると、「経済社会システムの中でのソフトウェアへの依存度はますます高まっており、我が国の情報サービス産業の 2003 年度の売上高は世界第 2 位の規模となっている。そのうち、ソフトウェア業務は 72.9% で、その中でも受注ソフトウェアが 81.9% を占めている。しかし、我が国のソフトウェア産業は、産学の連携が希薄であり、教育機関は産業界が求める教育を提供せず、産業界は大学における教育を採用の判断材料にしないといった負の循環が存在している。この状況を改善し、産業界と教育機関が連携していくことが必要である [1].」と述べられている。

大学におけるプログラミング教育の目的は、①プログラマ養成のためのコンピュータ言語教育、②問題解決方法の教育の 2 つに分類することができる [2]。ソフトウェア産業界が必要とする人材を育成するためには、学習したプログラミング言語を用いた②の教育が欠かせないと考えられる。すなわち、プログラム言語空間の中で、初期状態から目標状態へ到達するために、そのプログラミング言語を用いて思考を外化すること、それも構造的に思考を外化することを体験的に学習させるための教育である。

そのためには、例題のような断片的な学習ばかりではなく、現実に移動するソフトウェアを開発させる方法が最も有効であると思われる [3]。実際にソフトウェアを開発させることで、プログラミング教育のゴールを体験させ、それに至るプログラム理解の知識（プログラミング言語、変数、プロ

グラミング技法、アルゴリズム、バグの知識）とソフトウェアの構造・仕組みを設計するための知識を、自分自身の概念を形成しながら獲得させることが重要であると考ええる。

しかし、大学で取り上げられている言語は C や Java などのコード記述が主であり、一般に普及しているようなシステムやアプリケーションの開発には適していないこと、また、多人数を対象とする授業が多く、個々人のソフトウェア開発に対応することはできないことなどの理由で、実践的な教育はあまりされていないのが現状である。

開発現場では受注ソフトウェアのうち、GUI (Graphical User Interface) 対応のものが大半であり、特に VB (Visual Basic) を用いた案件がかなりの比重を占めているといわれている。VB は、ドラッグ & ドロップによりユーザインタフェースのデザインがしやすく、プログラムの実行過程を視覚的に確認できるため、比較的簡単にソフトウェアを作成することができ、開発コストが安くなるからである。

VB は、GUI 環境に対応したシステムやアプリケーションを作成するための言語として開発された。しかし、完全なオブジェクト指向言語ではない、実行速度が遅い、プログラミングの内部的な仕組みを理解できない、記述が冗長である等の問題を指摘され、大学のプログラミング教育のカリキュラムの中に取り入れることについては、まだ議論の最中である。しかし、オブジェクト指向言語として VB.NET が普及し始めてからは、Keyuan Jiang [4] のように、オブジェクト思考の概念の学習や、Web アプリケーションを作成させるために、カリキュラムに取り入れる大学も出始めている。今後、ソフトウェアの開発において GUI はますます重要な要素になってくると思われる。その際、GUI のオブジェク

トをコードのみで記述することは非効率的であり、GUI エディタを提供する開発環境の普及が予想される。

それに伴い、開発者には、従来からの「ソフトウェアの構造・仕組みの設計」に加えて、「GUI のデザイン」も要求されることになる。そして、グラフィカルな環境が普及すればするほど、どんなに優れたソフトウェアであっても、ユーザインタフェースのデザインが、ユーザーの感性やユーザビリティに合っていないければ、価値が下がってしまうという現象が起きる。また、ユーザインタフェースのデザインには、実際のソフトウェアの動きを評価しながらの修正が不可欠である。

2005 年に開催された第 7 回日本感性工学会大会の基調講演で、工業デザイナーの山中が、「今日のテクノロジーは、技術コンセプトの中に明快なライフスタイルのイメージが構築できていない。基礎研究段階からデザイナーを活用することが必要であり、これが研究の方向性に関わる重要な示唆をもたらす [5].」と述べている。基礎研究段階から技術者とデザイナーが共同作業を進める中で、逆に、技術者側にも、デザイナー的視点を持つことが求められるであろう。ソフトウェアについても同様と考えられ、これからのソフトウェア技術者には、ユーザーの感性も含めたユーザビリティに対する視点が必要になってくると思われる。

大学の授業の中でソフトウェアを開発する場合、「ソフトウェアの構造・仕組み」の設計者と、「GUI」のデザイナーが同一者であることが普通である。したがって、(1) プログラム理解、(2) ソフトウェアの構造・仕組み設計に加えて、(3) 「GUI のデザインとユーザビリティ」についても学習することとなり、これからのソフトウェア技術者に必要な視点を学ぶよい機会になると思われる。

一方、VB を用いたソフトウェアは、プログラムの実行過程を視覚的に確認しやすいため、他人が記述したコードを読むことが比較的容易である。そこで、アーカイブにソフトウェアと詳細なドキュメントを残すことで、前の学習者が習得した知識を、可能な限り後の学習者に継承することを考えた。

前述の (1) ~ (3) についての教育を目的とし、VB を用いた継承型の授業をデザインし、実践した結果について報告するが、(1)、(2) については、野村が [6] で述べている。本稿では、主に (3) 「GUI のデザインとユーザビリティ」の学習効果についての評価と考察を述べる。

2. VB について

2.1 に本授業で VB を用いる背景について、2.2 に VB の問題点に対する考え方について、述べる。

2.1 VB を用いる背景

本学の「情報メディア学科」は、文系・理系の複合学科である。アルゴリズムやプログラミングの学習など、情報系の科目で使用するプログラミング言語は、以前は C、2002 年度からは Java に統一している。現在は、学生が自宅でも学習で

きるように、無償提供されている Java2 SDK の MS-DOS コンソールを使用するタイプと、簡易な統合開発環境である Cpad for Java2 SDK を用いている。しかし、コードのみの Java を用いて現実のソフトウェアを開発するには、多くの時間がかかる。

そこで、少人数対象の VB を取り入れた授業を開設している。Visual C++ は基本的にはコード組み立て型であるため使用せず、1999 年度からは VB を、2003 年度からは VB.NET を、2006 年度からは VB2500 を学習している。Java にも JBuilder や Eclipse などの GUI エディタを提供する開発環境が普及してきたが、標準ツールが存在しないため、使用していない。

2.2 VB の問題点に対する考え方

最初に学ぶプログラミング言語として、VB が適しているとは思わない。記述のし方が冗長で、プログラミング言語の基本的な文法、データ構造、制御構造を学ぶためには適していない。最初はやはり、アルゴリズムの学習などに長い歴史を持つ手続き型の流れを汲む言語（現在なら C か Java）を選択すべきであろう。その上で、学科の状況に応じて、前述のプログラミング教育の目的②において、課題を設定しやすい言語を選択すればよいと考える。

本学科は工学部ではないため、卒業生は OS などの開発よりも、アプリケーションなどの開発部門へ進むことが多い。したがって、OS やプログラム言語処理系の学習に多くの時間を割くよりも、現実のソフトウェアの開発を行うことで、「プログラム理解」の知識、「ソフトウェアの構造・仕組み設計」の知識、「GUI のデザインとユーザビリティ」の知識を得ることの方が大切であると考え、そして、OS やプログラム言語処理系についての知識は、ソフトウェアを開発していく過程で必要になれば、ヘルプ機能などを用いて自ら学習することもある。（図 1 参照）

3. 授業デザイン

VB を学習する授業は 1999 年度から開設しているが、授業デザインを明確にしたのは 2002 年度からである。これは、

図 1 VB を用いた知識の獲得

表 1 情報領域カリキュラム

	前期	後期
1 年	情報システム論	コンピュータ基礎演習
	ソフトウェア論	情報数学論
	視覚情報論	デジタル表現論
2 年	システム設計論	システム演習
	プログラミング論	プログラミング演習
	情報数学基礎演習	アルゴリズム論
	データベース演習	データベース論
3 年		情報科学
	システムセキュリティ論	通信ネットワーク論
	プログラミング応用論	ネットワークプログラミング論
	情報数学応用論	情報数学応用演習
	コンピュータグラフィックス	パターン認識

教室の設備やソフトウェアの充実など、学習環境が整ってきたことが大きな要因である。3.1 に背景となる学習環境について、3.2 に授業デザインについて述べる。

3.1 学習環境

3.1.1 カリキュラム

本学科の情報領域の現行のカリキュラムは表 1 のとおりである。殆どが選択科目であるが、本授業の受講生は、ほとんどの科目を選択している。また、着色している授業で Java を使用しているが、“プログラミング～”と付加されている授業以外では、学習を進めるためのツールとしての使用が主である。カリキュラムにある「システム演習」は多人数対象の授業であり、実際のシステム開発は行われていない。

3.1.2 設備

授業を行う教室の設備は以下の通りである。

- ・ Windows PC (Dell OPTIPLEX GX240) …………… 13 台
- ・ 15 インチ液晶ディスプレイ (NANAOL365) …………… 13 台
- ・ プリントサーバ (Dell PowerEdge 500SC)
- ・ カラーレーザープリンタ (EPSON 2000C)
- ・ ネットワークスキャナ (EPSON ES-2200)
- ・ マルチメディア対応液晶プロジェクタ等 (三菱 LVP-XD200)
- ・ ネットワーク機器 (PLANEX FX-08MA)
- ・ ソフトウェア (Visual Studio.NET 等)

3.1.3 学習者

学習者の人数は 1 講座 8 ～ 12 名であり、学生の希望によって受講されるため、比較的プログラミング学習意欲の高い学生が集まっている。

3.2 授業デザイン

デザインした授業は、3 年次と 4 年次の連続した 2 年間 (4 コマ) にわたるものであり、図 3 に示すように、2 つのサークルを用いて表すことができる。これは、1999 年に発表された NRC の Commission on Behavioral and Social Sciences and Education による研究成果 “How People Learn” [7] を基に、Vanderbilt University で開発された学習者モデル “STAR Legacy modules” [8] (図 2 参照) を参考にしている。

図 2 STAR Legacy modules [9]

この学習者モデルは、学習サイクルを、①学習目的に基づく興味をひく課題やシナリオの提供 “CHALLENGE”、②課題に挑戦するために現在持っている知識の探索と確認 “THOUGHTS”、③新しい考え方や知識の獲得 “PERSPECTIVES & RESOURCES”、④評価 “ASSESSMENT”、⑤仕上げ “WRAP UP” の 5 段階で表したモデルで、様々な分野で学習支援システムを作成する際に利用されている。

本稿で提案する授業デザインは図 3 に示す 2 つの学習サークルから構成されている。白抜きの矢印で示す内側のサークルが 3 年次の授業に、着色した矢印で示す外側のサークルが 4 年次の授業に対応している。

3 年次の授業は教師により方向づけられており、講義と演習の形態を用いて VB の理解と習得を目的としている。3 年次の授業のスタートは、前年度の 4 年生が学習の最後に作成した「ソフトウェア作品」のデモンストレーションから始まる。これは、新しい学習者に現在保有している知識の探索を促すが、この段階では制作過程を想像することは不可能である。しかし、身近な学生が作成した、一般に市販され、普及しているようなシステムやアプリケーションの提示は、達成可能な目標を暗示し、新しい学習者に新しいプログラミング言語である VB を学習するための達成動機を喚起すると期待できる。

プログラミング学習は、本質的に演習型学習である。演習に適した例題の多いテキストを使用し、90 分の授業時間に 3、4 人の学生が交代で教師役を担当する Learning by teaching [10] 方式で学習する。プロジェクトに教師役学生のモニタ画面を表示して学習を進めるが、質問や異議があれば、その学生のモニタ画面に切り替えることができるネットワーク環境を利用した簡易な CSCL (Computer-Supported Collaborative Learning) [11] である。授業の準備中に理解できなければ、事前に教師に質問に来る。授業中は、新しい概念や重要な事項については教師が補足説明をする。新しい言語を 1 年間かけて学習し、図 3 に示す外側のサークルに向かう。

4 年次の授業は自己管理型学習であり、習得した言語の知識を用いて、自ら課題設定したシステムやアプリケーションを完成させる問題解決型学習である。これまでに学生が作成したソフトウェアを保管しているアーカイブにある作品を、作成したい作品のイメージを持って自ら探索し、参考にす

図3 学習サイクル

る。VBを用いたソフトウェアは、実現させたい機能を幾つかのフォームに分割して作成する。これらのフォームはプログラムの実行に従って表示されるため、視覚的に確認しやすい。通常、他人が記述したコードを読むことは容易ではないが、該当するフォームのコードを確認することは比較的容易である。アーカイブには完成したシステムやアプリケーションのソフトウェアを残すだけではなく、設計仕様書と取扱い説明書の詳細なドキュメントも作成させ、保管している。設計仕様書の主な内容は、ソフトウェア概要、プロジェクト構成、フォーム概要、プロシージャ概要、フローチャートで、

電子ファイルと印刷物の両方で残し、学生が自由に参照できる。図4にフォーム概要、表3にプロシージャ概要の例を示す。

企業が定型的に作成しているシステムではなく、学生が初めて作成するソフトウェアの開発は、プロトタイプを作成から始める方法が有効であると考える。簡単なプロトタイプを作成するか、前開発者の作品をプロトタイプとして、「自己評価」と「参考作品の参照」、「マニュアルやWebからの新しい知識の学習」を繰り返しながら開発を進める。ほぼ完成すれば、他者に評価を依頼し、可能な限り修正を行う。最後に設計仕様書、取扱い説明書を作成し、ソフトウェア作品と共にアーカイブに保管する。これらのソフトウェア作品が、

表2 プロシージャ概要の例

No	フォーム名 または オブジェクト名	プロシージャ名	機能
1	UserForm1	CommandButton1_Click	共通教育科目を UserForm3 の ListBox 2 ～ 5 に表示選択された条件に合ったシートをアクティブにする
2	UserForm2	CommandButton1_Click	選択された条件に合ったものを UserForm3 の TextBox1 ～ 30 に表示
3	UserForm2	CommandButton2_Click	UserForm1 の表示
4	UserForm3	CommandButton2_Click	UserForm3 の TextBox, ListBox を消去
5	UserForm3	CommandButton3_Click	TextBox1 ～ 30 のテキストを Userform4 の条件にあった ListBox1 ～ 6 に表示し、単位計算
6	UserForm3	CommandButton6_Click	ListBox2 ～ 5 で選択されているものを TextBox1 に移動
7	UserForm3	CommandButton6_DbClick	TextBox1 のテキストを ListBox1 に移動
8	UserForm3	CommandButton7_Click	ListBox2 ～ 5 で選択されているものを TextBox2 に移動
9	UserForm3	CommandButton7_DbClick	TextBox2 のテキストを ListBox1 に移動

図4 フォーム概要の例

次年度の3年生の授業のスタートに活用される継承型学習サークルである。

4. 評価と考察

この授業についての評価を、「プログラム理解」、「ソフトウェアの構造・仕組み設計」については、学生の反応とソフトウェアの作品から行った。詳細については[6]で述べているが、4.1、4.2に簡単にまとめる。また、本稿の目的である「GUI のデザインとユーザビリティ」の学習効果については、作成年度の異なるシステム5つを、同一時点で第三者に評価してもらうことで検証した。評価と考察を4.3に述べる。

4.1 「プログラム理解」についての評価と考察

VB.NET を学習してきた2005年度の4年生11名に、図3の授業デザインに示す内側の学習サークルを終えた3年次の終わりと、内側と外側の学習サークルを終えた4年次の終わりに、本授業に対するアンケートを実施した。また、内側の学習サークルを終えた2005年度の3年生11名にも同様のアンケートを実施した。

本学科では、本授業の受講生のみが2種類のプログラミング言語を学習するため負担が増えるが、70%の学生がこれを肯定している。30%の学生がVB.NETのみの学習を希望している。また、表1に示すように、授業内の学習時間はJavaの方が多い。そこで、どちらが理解しやすかったかという比較の形で質問をした。結果は以下のとおりである。

① プログラミング言語全体、変数についてはVB.NETの方が理解しやすい。

- ② プログラミング技法のうち、構造を伴うものについては、学習の初期ではシンプルなJavaの方が理解しやすい。しかし、現実のソフトウェアを作成し始めると、オブジェクトの動きと連動しているVB.NETの方が理解しやすい。
- ③ プログラミング技法のうち、関数プロシージャやサブプロシージャのような抽象度の高いものについては、フォーム上のツールオブジェクトの動きと連動させている例題が多いVB.NETの方が理解しやすい。
- ④ アルゴリズムについては、Javaの方が理解しやすい。
- ⑤ バグについては、VB.NETは、3年次は入力ミスや単純な構文エラーが多く、バグ検出と修正はかなりサポートされているため理解度が高い。しかし、4年次のソフトウェア作成の際には、実行時エラー、論理エラーが多くなり、デバッグ機能をうまく使いこなすまでには至っていないため、理解度が低くなる。

以上、プログラム理解については、全体としてはオブジェクトの動きを視覚的に確認できるVBが理解しやすいが、2種類の言語の特徴を使い分けて理解に役立てていることがわかった。

4.2 「ソフトウェアの構造・仕組み設計」についての評価と考察

アンケートの結果、実際にソフトウェアを作成することが、「ソフトウェアの構造・仕組み設計」と「GUI のデザイン」を理解するために、多いに役立ったという回答がほとんどであった。その際、最も参考になったものは、アーカイブのドキュメントであり、次にアーカイブのソフトウェアであった。

表3に、これまでに学生が作成したソフトウェア作品の仕様をまとめたものを示す。これは、企業でソフトウェアの

表3 学生が作成したソフトウェア作品

	年度	言語	種類	作品 No	ソフト名	フォーム	シート	プロシージャ	モジュール	メニュー	日付	Timer	保存	コード	人数
授業デザイン改訂前	2000	VB	ゲーム	0-1	まうず練習	10		47				○		618	3
		VBA	システム	0-2	実習室座席管理	5	8	15			○		○	292	2
	2001	VB	アプリ	1-1	近畿ガイド	2		26						450	2
		VB	アプリ	1-2	スケジュール帳	3		26		1 * 5	○		○	511	3
		VB	ゲーム	1-3	ジグソーパズル	5		54		1 * 4		○		876	2
		VB	ゲーム	1-4	オセロ	2		19	2			○		1260	1
	2002	VBA	システム	2-1	就職部相談予約	13		16			○		○	510	2
		VB	ゲーム	2-2	マウス練習	9		36				○		270	2
		VB	アプリ	2-3	サウンドノベル作成	1		20	3	1 * 2			○	300	1
		VB	アプリ	2-4	グラフィックエディタ	11		60	2	4 * 3				750	1
授業デザイン改訂後		VB	ゲーム	2-5	Flipper	6		19				○		300	2
	2003	VBA	システム	3-1	単位取得手引き	12	13	68						6162	4
		VB	ゲーム	3-2	マウスラーニング	8		153	1			○	○	4788	1
		VB	ゲーム	3-3	うさ子	15		46				○		450	3
		VB	ゲーム	3-4	神経衰弱	23		46				○		360	2
		VB.NET	Web アプリ	3-5	スケジュール帳	7		17		2 * 2	○		○	420	2
	2004	VBA	システム	4-1	履修登録	7	16	58						2814	3
		VB.NET	ゲーム	4-2	シルバー世代マウス練習	9		115				○		1344	2
		VB.NET	Web アプリ	4-3	スケジュール帳	8		43	1	2 * 2	○		○	1836	2
		VB.NET	ゲーム	4-4	地理学習	10		50	1	1 * 4, 5 * 2	○	○	○	924	2
	2005	VBA	システム	5-1	教室予約	20	17	90	2		○		○	12960	4
		VB.NET	アプリ	5-2	就職面接	32		10						1656	1
		VB.NET	ゲーム	5-3	英単語学習	16		61	1			○		2160	2
		VB.NET	ゲーム	5-4	漢字学習	13		33		1 * 3 * 3	○	○	○	1404	2
		VB.NET	システム	5-5	キャリアセンター相談予約	19		165	1	1 * 3 * 4	○		○	11016	3

注) *メニュー : メニュー数*アイテム数*メニューの個数

*保存 : ○履歴や結果のファイルへの保存

*日付, Timer : ○システムの機能使用

*コード : プログラムコード行数

見積もりを行う際に使用される、ファンクションポイント法（FP法）[12]というソフトウェア評価法の評価項目を参考にしたものである。フォーム数、プロシージャ数、コード数も年々増加しており、メニューの作成、日付やTimer機能の活用、必要データのファイルへの保存など、作品の機能が充実していく傾向が認められる。継承型学習の効果があると推測される。また、特に授業デザイン改訂後の作品の向上が認められる。

4.3 「GUIのデザインとユーザビリティ」についての評価と考察

4.3.1 「GUIのデザインとユーザビリティ」評価法

ISOによると、コンピュータシステムのユーザビリティとは、「ユーザとコンピュータシステムのインタラクションの質であり、『ある環境において特定のユーザが特定の目標を達成する際の、効果、効率、満足の度合い』である」と定義されている[13]。Nielsenは、ユーザビリティのために行うユーザーインターフェース・デザインの体系的な診断方法として、10項目からなる「ヒューリスティック評価法」[14]を提案している。また、黒須らは、これを改良した「構造化ヒューリスティック評価法」[15]を提案している。

「ヒューリスティック評価法」、「構造化ヒューリスティック評価法」共に、「問題点を発見」する手法であり、評価の専門家が3人～5人いれば妥当な結果が得られると言われている。本研究では、作成されたソフトウェアのユーザビリティを、専門的、厳密に評価することが目的ではなく、VBを用いた継承型学習が「GUIのデザインとユーザビリティ」の学習に効果があることを検証することが目的である。

4.3.2 「GUIのデザインとユーザビリティ」の相対的評価

ソフトウェアのユーザビリティについて学習する「ユーザビリティエンジニアリング」の授業は、本学科では特に設けてはいないが、プロトタイプを使用する方法が有効であると言われている[14]。

通常学生が作成したソフトウェアは、3～5人の第3者に簡単なユーザビリティの評価を依頼し、評価結果をドキュメ

ントに残している。そこで、本授業の受講生が、これらを参考に、また、アーカイブのソフトウェアそのものをプロトタイプとして実行させることにより、ユーザビリティについて学習していることを検証したい。

学生がこれまでに作成したソフトウェアの継承関係は、図5に示すとおりである。作成時期の異なる複数のソフトウェアを、同一時点で同一基準を用いて第三者に評価してもらうことで、「GUIのデザインとユーザビリティ」の相対的な評価は可能であると考えられる。

「ヒューリスティック評価法」は「問題点を発見」するための技法であり、「定量化して評価」する技法ではない。しかし、評価基準を定めるに当たり、Nielsenの「ヒューリスティック評価法」の評価項目を参考にチェックリストを作成した。そして、異なるソフトウェア間の「GUIのデザインとユーザビリティ」を相対的に比較するために、5段階の間隔尺度を用いた。間隔尺度を用いるには、「構造化ヒューリスティック評価法」の方が適切であると考えられている。しかし、チェック項目が多く、同時点で複数ソフトウェアの評価テストを行う場合は評価者の疲労が大きく、今回のケースには適していないと思われる。

今回は、図5の右端着色部で示した大学内事務処理システムに関するものを用いて、評価テストを行った。学生によって開発されたシステムは、大学内で学生が現実に使用可能なシステムである。メンテナンスの問題があり、現在は実際に稼動しているシステムはないが、4年生であれば、就職部（現キャリアセンター）のシステムも含めて、システムの機能について理解することができる。さらに、「単位取得手引き」と「履修登録」システムは、2005年度から稼動している学内の統合的な教務処理システムに先がけて開発され、参考システムとなったものである。そこで、システム設計の知識もあり、ドメインの機能についても理解している、本授業を受講している現4年生を評価者とした。評価テストの方法は以下のとおりである。

- ①目標 : 「GUIのデザインとユーザビリティ」学習効果の確認
- ②評価者 : 本授業受講の4年生 10名
- ③試行時期 : VBを1年半学習した後

図5：ソフトウェアの継承

④試行時間：午前 10:00～12:00（疲労の少ない時間）

⑤試行場所：コンピュータ実習室

⑥方法：
・5つのシステムをパソコンにインストール
・5つのシステムの「取扱説明書」を用意
・評価項目チェックリストを用意（5段階評価）
・システムの作成年度は非公開，試行順序は自由
・テスト中の質問は可

4.3.3 「GUI のデザインとユーザビリティ」学習効果の評価と考察

評価テスト中の質問は，特に無かった。

表4の「項番」10までに表示している「項目」が，Nielsenの「ヒューリスティック評価法」の評価項目を参考にしたものである。学生が作成したシステムは「ショートカット」と「ヘルプ」を作成していないので省き，Nielsenの詳細項目を参照に，より具体的な10の「評価項目」を設定した。また，「項目」にある「11. システムの難易度」と「12. シス

テムの充足度」は，作成年度の違うシステムを評価者がどう捉えているのかを知るために追加したものである。

評価者には作品の作成年度を明示していないし，評価する順番についても指示していない。表4，図6に10人の評価者の平均値を示す。4.2に示したように，「ソフトウェアの構造・仕組み設計」については，アーカイブのドキュメントとソフトウェアからの継承型学習の効果が認められ，年度を重ねる毎にシステムの難易度は高くなっている。評価者はこれを正確に捉え，システムの充足度についても適切な評価をしていると考えられる。

評価項目の③以降については，後で開発された作品の評価が概ね高く，各々の開発者がアーカイブのドキュメントを参考に，また，アーカイブのソフトウェアを実行させることにより，ユーザビリティについて学習していることを推測させる。システムの難易度が高くなれば，ユーザビリティへの配慮まで手が回りにくいという懸念を持ったが，システムの難易度，システムの充足度と評価項目の③以降には0.7以上の

表4 「GUI のデザインとユーザビリティ」の相対的評価テスト結果

項番	項目	開発年度					⑪との相関	⑫との相関
		評価項目	2000 実習室座席管理	2002 就職部相談予約	2003 単位取得手引き	2004 履修登録	2005 キャリアセンター予約	
1	シンプルで自然な対話	① グラフィックデザインとカラー	3.1	3.4	4.3	4.1	3.4	0.21
		② 1画面の情報は適当か	3.7	3.0	3.7	4.0	3.3	-0.17
2	ユーザーの言葉で	③ 操作のわかりやすさ	2.8	2.8	3.2	3.4	3.3	0.73
3	ユーザーの記憶負荷を最小限に	④ 前画面の情報の記憶は必要か	3.3	3.1	3.3	3.7	4.0	0.86
4	一貫性	⑤ 操作、言葉に一貫性はあるか	3.8	3.7	3.7	3.7	4.0	0.63
5	フィードバック	⑥ 操作後のフィードバックは適切か	2.6	3.1	3.0	3.7	4.3	0.98
6	出口を明らかに	⑦ 非常出口はあるか	2.7	2.6	2.8	2.7	3.3	0.83
7	ショートカット							
8	適切なエラーメッセージ	⑧ 適切なエラーメッセージはあるか	2.3	2.6	2.6	3.1	3.7	0.98
9	エラーの回避	⑨ エラーの回避機能はあるか	2.6	2.6	2.4	3.1	3.3	0.81
10	ヘルプとドキュメンテーション							
		⑩ 取説はわかりやすいか	3.1	3.1	3.3	3.5	3.6	0.89
11	システムの難易度	⑪ システムの難易度	2.8	3.3	3.4	3.7	4.4	
12	システムの充足度	⑫ システムの充足度	2.9	3.1	3.4	3.6	4.1	0.98

図6 「GUI のデザインとユーザビリティ」の相対的評価テスト結果

高い相関がみられ、継承型学習の効果が認められた。

「単位取得手引き」は「履修登録」を、「就職部相談予約」は「キャリアセンター予約」を継承しており、これらの評価結果をみると、ユーザビリティ学習の効果が大きいことがわかる。具体例として、「単位取得手引き」では、ユーザーのエラー操作に対するエラーメッセージが少ないが、これを実行させることにより、「履修登録」では、予測できるエラーに対してメッセージを表示し、エラーを回避するように工夫している。

一方、①グラフィックデザインとカラー、②1画面の情報量の適切さ、の2項目については、連続した学習効果はみられない。①グラフィックデザインとカラー については、開発者の感性、好みが強く反映され、学習する対象とはなりにくいと思われる。しかし、今回の評価の目的ではないが、評価者の評価が高いものは、機能に重点を置いた無機質な画面よりも、カラーを効果的に使用しているものであることがわかった。また、②1画面の情報量の適切さについても、開発者の感性が優先されているようだが、評価結果からは、ユーザーに inputs を要求する画面を少なくし、1度に大量のデータを入力させる方法は、好まれないことがわかった。この方法は、ユーザーの記憶負荷を減らすためには効果的であるが、ユーザーはコンピュータとの対話を求めているようにもみえる。以上、①、②の項目については、アーカイブズからの学習効果は少ないと思われる。しかし、今後、今回行った評価結果を明示的に提示することで、学習させることが可能であると考えられる。

5. お わ り に

5.1 ま と め

身近な題材を基に、現実に稼動するソフトウェアを開発させることでプログラミング教育のゴールを体験させ、それに至る(1)プログラム理解、(2)ソフトウェアの構造・仕組み設計の知識を自分自身の概念を形成しながら獲得させていくことは重要であると考えます。

これを実現するために、認知的負荷の少ないVBを用いた。VBを用いたソフトウェアは、プログラムの実行過程を視覚的に確認しやすいため、他人が記述したコードを読むことが比較的容易である。そこで、アーカイブにソフトウェアと詳細なドキュメントを残すことで、前の学習者が習得した知識を、可能な限り後の学習者に伝える継承型の授業をデザインして実施した。

(1)、(2)については学習効果が得られたことを別稿で述べている。本稿では、この継承型学習が、(3)「GUIのデザインとユーザビリティ」の学習にも効果があることを検証した。すなわち、作成時期の異なる複数のソフトウェアを、同一時点で同一基準を用いて第三者に評価してもらうことで、「GUIのデザインとユーザビリティ」の相対的な評価を行い、学習効果を確認した。その結果、継承型学習の効果が認められた。また、以下の「GUIのデザインとユーザビリティ」に対する知見も得られた。

- ① 機能重視の画面より、カラーを効果的に使用した画面デザインが好まれる。
- ② 1度に大量のデータを入力させる方法は好まれない。
- ③ ユーザーはコンピュータとの対話を求めている。

5.2 今 後 の 課 題

今回の「GUIのデザインとユーザビリティ」の相対的な評価テストは、大学内事務処理システムのみを対象としたが、今後はゲームや一般的なアプリケーションについても評価を行い、学習の継承を検証していきたい。

また、今回の主目的ではなかったが、「GUIのデザインとユーザビリティ」に対する知見も得られたので、今後の学習に活かしたい。

エラー回避機能の中で、ユーザーが行った操作を1つ前の状態に戻したり、取り消したりすることについては、プログラムの状態遷移、変数保存のタイミングなどの緻密な把握が必要で、簡単に対応できないものもある。今後、これらについて知ること、対応することも大事な学習であると考えている。

参 考 文 献

- [1] 経済産業省、厚生労働省、文部科学省：平成15年度ものづくり白書（製造基盤白書）、pp. 174-176, 2005
- [2] 家本修：プログラミング教育、教育工学事典、459, 2000
- [3] 野村典子：現実のシステムを開発・運用するインタラクティブな環境がもたらす学習効果、日本教育工学会論文誌、vol.27, Suppl, pp.205-208, 2003
- [4] Keyuan Jiang et al.: A different approach of teaching introductory visual basic course, Conference On Information Technology Education, Proc. of the 5th conference on Information technology education, Salt Lake city, UT, USA pp.219-223, 2004
- [5] 山中俊治：美しいテクノロジーをめざして－研究開発におけるデザインと美的感覚の役割、第7回日本感性工学会大会予稿集 2005, 31, 2005
- [6] 野村典子：Visual系言語を用いて現実のソフトウェアを開発する循環型学習サークルのデザイン、教育システム情報学会誌、Vol.24, No3, 2007（掲載予定）
- [7] M.Suzanne Donovan, John D.Bransford, and James W.Pellegrino: "How People Learn", NATIONAL ACADEMY PRESS, Washington,D.C., 1999
- [8] Sean Brophy, John Bransford: "Design Methods for Instructional Modules in Bioengineering", Proc. of the 2001 American Society for Engineering Education Annual Conference & Exposition, 2001
- [9] <http://iris.peabody.vanderbilt.edu/slm.html>
- [10] Michie, D., A. Paterson, et al.: "Learning by Teaching", Second Scandinavian Conference on Artificial Intelligence (SCAI), Finland, 1989

- [11] 稲葉晶子：CSCL, 教育システム情報ハンドブック, 実教出版, pp.91-94, 2001
- [12] David Garmus and David Herron (小泉浩 他 (訳))：ファンクションポイントの計測と分析, ピアソンエデュケーション, 2002
- [13] 加藤秀一：ユーザビリティの概念, ヒューマンインタフェース, オーム社出版局, pp.352-354, 1998
- [14] Jacob Nielsen：Usability Engineering (三好かおる (訳))：ユーザビリティエンジニアリング原論, 東京電機大学出版局, 1999
- [15] 黒須正明, 杉崎昌盛, 松浦幸代：問題発見効率の高いユーザビリティ評価法 - 1. 構造化ヒューリスティック評価法の提案 -, ヒューマンインタフェースシンポジウム #13, pp.481-488, 1997

野村 典子 (正会員)

1971 年 日立製作所情報システム研究所入社。1994 年 武庫川女子大学・生活環境学部・生活情報学科・専任講師。1998 年 武庫川女子大学・生活環境学部・情報メディア学科・助教授, 2007 年から准教授。高速計測データを用いた学習における認知過程, プログラミング教育の研究に従事。情報処理学会, 日本感性工学会, 教育システム情報学会, 日本教育工学会, ゲーム学会各会員。

太田 健一 (正会員)

工学博士。1978 年 兵庫県立工業技術センター主任研究員。1993 年 兵庫県立大学・工・情報工学科助教授, 2001 年 武庫川女子大学・環境・情報メディア学科教授, 現在に至る。色彩画像処理, CAD 開発などに従事。日本感性工学会, 繊維機械学会等会員。