

携帯電話用組み込みソフトウェア開発の実践的教育における 産学連携の課題

伊藤 篤[†] 渡辺 裕^{††} 樽松 明^{†††}
 村上 仁己[†] 湯山 一郎^{††} 宮道 壽一^{††}
 長谷川 光司^{††} 細島 美智子^{††}
 月川 淳^{††} 藤井 雅弘^{††}

今後、コピキタス社会の進展にともない各種の機器にプロセッサが組み込まれる。このコピキタス社会を日本がリードするためには、各種機器に組み込まれたソフトウェアの効率的な技術開発力の強化が必須となる。このためには、コーディングだけでなく、企画立案から製品化までをこなす、実際的なスキルを備えた有能な人材を幅広く多数育成する必要がある。ここで、組み込みソフトウェア技術については、設計段階までかわりを持つ総合化技術となるため、その教育は、大学などの高等教育機関を中心に実施し、実際のシステムにかかわる問題解決のスキルの習得については、産業界から最新の手法を取り入れ、実習教材や実習カリキュラムなどを新たに検討する必要がある。このような現実的な課題を解決し、企業の要求と大学教育のギャップを埋めるための教育手法の開発と、その継続可能性の評価を目的とし、「平成17年度経済産業省委託産学協同実践的IT教育促進事業」の一環として、KDDI、宇都宮大学、実務能力認定機構が協力し、携帯電話アプリケーションに関するIT教育実験を実施した。その成果を紹介するとともに、これをベースに、産学連携によるIT教育実施における問題点とその解決方法を評価する。さらに、今後の課題を明確化し、IT技術教育を充実させ、人材を早期に育成するために実施すべき施策の提言を行う。

Issues on Business-academia Collaboration in the Development of Educational Course; For the Design, Development and Implementation of Practical Cell Phone Software

ATSUSHI ITO,[†] YU WATANABE,^{††} AKIRA KUREMATSU,^{†††}
 HITOMI MURAKAMI,[†] ICHIRO YUYAMA,^{††} JUICHI MIYAMICHI,^{††}
 HIROSHI HASEGAWA,^{††} MICHIKO HOSOSHIMA,^{††}
 ATSUSHI TSUKIKAWA^{††} and MASAHIRO FUJII^{††}

In the future ubiquitous network society, thousands of highly distributed and interconnected appliances are equipped with microprocessors. Japan is in a leading position towards the realization of a ubiquitous network society. In order to maintain and extend the Japan's strengths in this arena, it is essential to develop Japanese engineer's software production faculties, including the ability to make plans for services realized through processor equipped appliances. Business-academia collaboration for it education on the nurturing of talented people for appliances software developments is expected to actively address various issues for the realization of a ubiquitous network society. However, business-academia collaborations for it education may pose new problems on costs, systems and regulations. To solve the problems, KDDI, a Japan's major telecommunications operator, in cooperation with Utsunomiya University and ACPA (Accreditation Council for Practical Abilities) demonstrated business-academia collaboration to develop educational courses for cell phone application software developments. This paper focuses on the results obtained through the educational course, and highlights new challenges of business-academia collaborations in education. We show that collaborative efforts between enterprises and universities with supports from government are particularly important to solve the problems and to improve effectiveness of IT education.

[†] KDDI 株式会社
 KDDI CORPORATION
^{††} 宇都宮大学
 Utsunomiya University

^{†††} 特定非営利活動法人実務能力認定機構
 Accreditation Council for Practical Abilities

1. はじめに

今後、ユビキタス社会の進展にともない各種の機器にプロセッサが組み込まれる。このユビキタス社会を日本がリードするためには、各種機器に組み込まれたソフトウェアの効率的な技術開発力の強化が必須となる。このためには、実際のスキルを備えた有能な人材を幅広く多量育成する必要がある。経団連の答申にもあるように、企業は、大学にはないビジネスの実環境に根ざした実務知識・ノウハウの提供を行うべく、ソフトウェアや開発ツールを含めた教材の提供、第一線で活躍する企業人の教員派遣など、積極的な協力を行う必要がある。企業からの協力のもと、大学は、企業や海外の先進的教育機関の教育プログラム、教育手法、教材、教員などを積極的に採用し、専門教育を実施する。ここで、講座設計、運営にあたっては、NPO実務能力認定機構のスキル評価、単位認定制度や、経済産業省のITスキル標準などの積極的な活用が有効である¹⁾。

特に、組み込みソフトウェア技術は、設計段階までかわりを持つ総合化技術となるため、その人材の育成は、大学などの高等教育機関が中心となり、実際のシステムにかかわる問題解決のスキルの習得については、産業界から最新の手法を取り入れた実習教材や実習カリキュラムなどの面で協力を得て強力に進める必要がある²⁾。

組み込みソフトウェアの中で、携帯電話用のソフトウェアは、現在、エンタテインメント系の利用が中心であるが、今後は、教育や健康管理など、ユビキタス社会におけるコアデバイスとして重要な役割を果たすものと考えられる。携帯電話のアプリケーションの開発言語については、従来から利用されてきたJAVAに加え、携帯電話の機能を自由にコントロールでき、起動や処理が高速なBREW(Binary Runtime Environment for Wireless)³⁾も普及し、より、高速かつ高機能なアプリケーションの開発が可能となっている。

本稿では、BREWを利用した携帯アプリケーションの製作に注目し、その開発技術の教育を行った結果得られた知見を述べる。BREWを利用した携帯電話用の組み込みソフトウェアは、WEB的な画面表示とイベントドリブンな状態遷移という2つの異なる要求を効率良く満たすようなプログラムの開発が必要であるが、この両方の開発に精通した技術者は現状きわめて限られている。また、同じBREWを利用した携帯電話でもメーカーによりプラットフォームは異なっているという現実があり、機種ごとのコーディングと試験

も必要となる。

また、携帯電話アプリケーションの場合、アプリケーション完成後、商用に至るまでには、ビジネス企画、機種ごとの全パスをトレースする検証が行われ、膨大な分量のドキュメントを作成することが求められる。このため、ビジネスを明確に企画し、システムティックに、画面遷移・画面設計・状態遷移表などの中間のドキュメントを作成しながらアプリケーション開発し、検収試験を実施できるという、企画立案から最終製品化までをこなすスキルが、コーディング技術のほかに求められている。

そこで、このような現実的な課題を解決し、企業の要求と大学教育のギャップを埋めるための教育手法の開発と、その継続可能性の評価を目的とし、KDDI、宇都宮大学、実務能力認定機構⁴⁾が協力し、携帯電話アプリケーションに関するIT教育実験を実施した。この教育実験は、「平成17年度経済産業省委託産学協同実践的IT教育促進事業」⁵⁾の一環として行われ、産学連携によるIT教育実施における問題点とその解決方法についての評価と今後の課題を明確化した。

まず、2章では、今回実施した教育訓練プログラムの内容を概観し、課題設定と本教育訓練の特色を述べる。3章では、学生へのアンケートと講師による成果物の評価結果をベースにして教育訓練プログラムの成果を評価する。4章には今回のプログラムの実施で明らかとなった産学連携による教育の課題と、解決手段を述べる。5章では、大学やインストラクタからみた本実験のIT教育展開に向けた評価を述べ、6章で今後の産学協同IT教育への提言を行う。

2. 実施内容

今回実施した教育訓練プログラムの構造を図1に示す。大学側は、学生にC++などの計算機言語の知識を与え、本教育訓練プログラムを受けられる状態にするとともに、教育訓練環境(コンピュータ教室、VISUAL C++や.NET)を提供し、授業時間外の学生からの質問の一次受付などのサポータとしての役割を果たした。これに対して、企業側は、主に、BREW開発環境(コンパイラや携帯電話)の提供、資料・教材の提供、サンプルコードの提供、講師の提供、などを行った。実務能力認定機構は、職種と職務に必要なスキルとその到達目標を体系的に定めるノウハウを利用し、成果の評価を実施した。

教育訓練は、平成17年10月7日~12月16日までの全10回で実施したが、そこでは、以下の3課題を設定した。

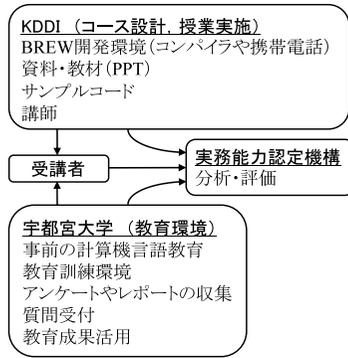


図 1 教育訓練プログラムの構造

Fig. 1 Players and their roles of the training course.

- (1) 組み込みソフトウェアの一例としての、携帯電話アプリケーションを作成できるようになること
- (2) 企画書、仕様書、状態遷移図など、ソフトウェア開発にかかわるドキュメントを作成できるようになること
- (3) グループでのソフトウェア開発ができるようになること

これらに基づき、以下のようにコース設計を行った。1~4回は講義形式とし、携帯電話のネットワーク、ハンドセットの構造、BREWのアーキテクチャ、BREW APIを学び、5~10回はアプリケーション作成実習を行い、BREWを利用した携帯電話アプリケーション作成スキルを身に付けた。実習に入ってから、グループ演習を取り入れ、作業分担によるリーダーシップの向上を目指した。

携帯電話向けの組み込みソフトウェア開発を題材とした、本教育訓練の特色を以下に述べる。

組み込みソフトウェア開発に特有な環境での教育

組み込みソフトウェア開発では、PCなどの汎用系のシステム上でターゲットシステムのプログラム開発を行う、クロス開発が行われる。組み込みソフトウェア開発を行うためには、これらの開発のためのツールを使いこなすスキルが必須である。また、テストや動作確認を行うためには、エミュレータを使用する。今回のプロジェクトでは、エミュレータの使用環境を実際に使用してその機能を理解することに加えて、デバッガを使用して実習で作成するアプリケーションソフトウェアのデバッグ方法を体験し、その方法も習得した。BREWについては、分かりやすい日本語の参考書⁶⁾も出版されているため、学生の実習環境として適当であるといえる。

開発プロセスのエンジニアリング化

一般的に、組み込みソフトウェアにおいてミスが起

こった場合、その影響が甚大であることが多い。そのため、組み込みソフトウェアには、エラーのないプログラムが要求され、企業では品質を保證できる開発体制が敷かれている。組み込みソフトウェアを職人技術に依存しない製品として製作するためには、開発プロセスの標準化と要求事項の明確なドキュメント化とテストによる検証が必要となる。組み込みソフトウェア開発のプロセスは、大きく分けて次の4つに分けられる。

- (1) 仕様書に基づく要求分析
- (2) 要求仕様に基づくソフトウェア設計
- (3) ソフトウェア設計書によるプログラム設計、プログラム作成、単体テスト
- (4) テスト仕様書に基づくプログラムテスト

本教育訓練では、このような開発プロセスを、演習を通じて体験することで、開発のプロセスを具体的に理解し、仕様作成からプログラム作成を経てテストに至るまでの、一貫した開発スキルを習得することができる。また、グループのメンバでソフトウェアを共同制作することにより、各種ドキュメントを作成し、グループメンバの間でのコミュニケーションを確実に行うスキルが習得できることを目的とした。

受講者サポート体制

本教育訓練開始時に、趣旨などについてのオリエンテーションを実施した。また、実施中は随時電子メールによる質問を受け付けた。

3. 実施結果

3.1 学生へのアンケートからの総合的な評価

アンケートによる学生の習得についての評価をまとめると以下ようになる。おおむね良好な評価で、この教育訓練は有効であったと判断できる。

- ソフトウェア開発の能力の向上がみられた。大学院修士1年学生でありC言語のプログラミング能力をそなえていたことから、スムーズにアプリケーションソフトウェアの作成ができた。
- ソフトウェア開発に必要な、計画書作成、仕様定義、テスト項目作成などの作業を実施できた。
- 実行計画作成、進行管理などプロジェクトマネジメントは、これまで大学のカリキュラムにはないものであったが、このプロジェクトにより実施できて、その基本的スキルの習得ができた。
- 携帯電話用アプリケーションソフトウェア開発の仕事に興味をおぼえたので、今後発展させたい。

3.2 理解度についての評価

理解度は、授業後のアンケートにおいて、学生が5段階評価で評価した。毎回の講義における理解度の

推移を表 1 に示す．ここで「平均理解度」とは，理解度の評点の平均値である．また，「理解率」は，評点 3 以上の人数の受講生全体に対する割合である．次に，BREW プログラミングの平均理解度と理解率を表 2 に示す．なお，学生が作成した BREW のアプリケーションを，KDDI から派遣された講師やインストラクタにより評価した結果，誤りもあったものの，アプリケーションが動作したことで最低限の品質はクリアしていた．

3.3 開始前と修了時の項目別理解度の比較

プログラミングに対する理解，技術力，プログラム開発手順，ならびに組み込みソフトウェアに関する理解に関連する各項目に対する「平均理解度」と「理解率」の差を調べた．その結果を表 3 に示す．「C プログラミング」に関する講義開始前の自己申告による平均理解度は 3.7 であるが，終了後の平均理解度は 3.0 となっている．また，「プログラム開発手順」については，講義開始前の自己申告による平均理解度は 4.1 であるが，終了後の平均理解度は 3.5 となっている．この 2 項目は開始前よりも修了後の方が下がっているが，これは，実際にアプリケーションを作成してみて，プログラミング能力・プログラム開発手順に関する能力の不足に気がついたといえる．これに対し，組み込みソフトウェア，組み込みソフトウェア開発環境，プログラムデバッグの項目においては，理解度が向上し，スキルが習得できたことがうかがえる．また，評価 3 以上の人数の割合を示す理解率でみると，講義開始前の C プログラミングの理解率は 92% と高かったが，これは，前述のように，思い込みによる数値であ

り，修了時には，79% と下がっている．組み込みソフトウェアについては，開始前は理解率が 17% でほとんど理解してなかったが，終了時には 42% となり向上していた．これは，理解率は向上しているが，まだこれだけの実習では，組み込みソフトウェア開発のスキルの達成には十分とはいえないといえる．組み込みソフトウェア開発環境は，開始前は低かったが，終了時は 70% 程度まであがっていた．プログラムデバッグは，開始前が 67%，終了時が 75% であり向上がみられた．

3.4 スキル要素の習得度

携帯電話用アプリケーションソフトウェアの開発に関わる詳細なスキル要素について，修了時のスキル習得度を調べた．習得度の尺度も，理解度と同じ 5 段階である．その結果の一部を表 4 に示す．詳細は文献 5)

表 3 項目別理解度（開始前と修了時の差）

Table 3 Attainment levels of programming skills (before and after taking the training course).

項目	質問項目	平均理解度	理解率
C プログラミング	開始前に C 言語基礎を理解していた（自己申告）	3.7	0.92
	修了後，携帯電話プログラミングの能力が向上した（アンケート）	3.0	0.79
プログラム開発手順	開始前に C 言語プログラム開発手順を理解していた（自己申告）	4.1	0.92
	修了後，携帯電話プログラムの開発手順が理解できた（アンケート）	3.5	0.92
組み込みソフトウェア	開始前に組み込みソフトウェアを理解していた（自己申告）	1.6	0.17
	修了後，組み込みソフトウェアを理解した（アンケート）	2.6	0.42
組み込みソフトウェア開発環境	開始前に組み込みソフトウェアの開発環境を理解し実行できた（自己申告）	1.6	0.08
	組み込みソフトウェアの開発環境を理解し実行できた（アンケート）	2.9	0.71
デバッグ	開始前にデバッグを使い C++ プログラムのデバッグができた（自己申告）	3.0	0.67
	C++ と BREW プログラムのデバッグを習得した（アンケート）	3.3	0.75

表 1 毎回の授業の理解度

Table 1 Understanding level of each lecture.

日付	平均理解度	理解率	5	4	3	2	1
10/14	3.2						
10/21	2.8						
10/28	3.5						
11/11	3.8	0.88	5	8	8	1	0
11/18	3.3	0.79	2	10	7	3	2
11/25	3.5	0.83	4	8	8	4	0
12/ 2	3.4	0.85	3	10	10	3	1
12/ 9	4.2	1.00	8	12	3	0	0
12/16	3.4	0.83	2	10	8	4	0

表 2 BREW プログラミング理解度

Table 2 Attainment level of overall BREW programming skills.

日付	平均理解度	理解率	5	4	3	2	1
12/ 9	3.6	0.77	3	10	7	3	0
12/16	3.3	0.88	2	7	12	3	0

を参照されたい。この結果から、組み込みソフトウェアの理解とイベントハンドラとコールバックの処理によるプログラムの作成が、十分とはいえないことが示された。これは、これまで経験したことがないスキルであるためであると考えられる。経験がないものを学習するには、より多くの時間をかけることが必要であるが、今回の講義の中では、時間の制約からこれらのスキルの習得に十分な時間をかけられなかったという反省がある。

3.5 講師による評価

学生のアンケートによる評価だけでは不十分である。そこで、講師が、学生が作成したアプリケーションおよび仕様書などのドキュメントを評価した結果、C言

表 4 修了時のスキル習得度評価
Table 4 Overall attainment levels after taking the training course.

質問項目	平均評価値	理解(達成)率
イベントハンドラとコールバックの処理によりプログラムを作成することができるようになった	2.3	0.58
C++ソースコード、BREWソースコードを作成できるようになった	2.9	0.75
組み込みソフトウェアの概念を理解できるようになった	2.9	0.58
ソフトウェア開発手順を理解できるようになった	3.4	0.83
課題について環境を把握し、制約事項を整理して分析ができるようになった	2.9	0.63
コールバックが理解できるようになった	2.8	0.58
ユーザインタフェースの設計ができるようになった	2.9	0.63
詳細設計としてデータ構造の作成ができるようになった	2.8	0.67
詳細設計として画面遷移とイベントの関係を理解できるようになった	3.7	0.92
状態遷移図が作成できるようになった	3.8	0.95
テスト項目を作成し、完了基準を設定できるようになった	2.9	0.75
企画を立て、企画書を記述できるようになった	3.3	0.92
作業項目を整理しその進捗をチェックできるようになった	3.4	0.79
ソフトウェアを開発するのに必要なドキュメントの読み方ができるようになった	3.5	0.92
プロジェクトの実行に関してチーム内でコミュニケーションがとれるようになった	3.6	0.75

語の基本的な理解の不足が目立った。たとえば、以下の評価が講師から出たが、致命的な問題点はなかった。

- サンプルのソースコードをトレースする力が不足している。もうすこし、しっかりとソースコードをトレースしていれば回避できた問題が多かった。
- They didn't handle the case properly where the write to the temporary file fails.
- Students should pick up a book on C and read up on the basics of pointers, malloc, and FREE in order to understand why this code is incorrect.
- This stylistic but when working on existing code, it's generally a good idea to follow the existing coding conventions.

また、ドキュメントとしては、企画書、画面遷移図、画面詳細設計図、状態遷移表、テスト仕様、進捗管理表を作成させた。講師の目から見た、課題のサンプルコードから想定される記述内容を100%とし、提出されたドキュメントとの比較を行った結果を表5に示す。なお、表中の完成度は、各ドキュメントの完成度を平均したものである。

ドキュメント作成に関する戸惑いがあり、また、アプリケーションに関する理解度が低いため、中盤過ぎまでは、エラーシーケンスが抜けていたり、状態遷移の書き方そのものがおかしかったりしたが、後半になって理解が進むと、100%の品質まで達することはできなかったものの、急激に完成度が向上した。

これらのことから、本教育訓練を通じて実践的な経験を踏むことにより、プログラミングやドキュメント作成についての理解を深めることができたといえる。

表 5 ドキュメントの完成度
Table 5 Qualities in documentations.

回	授業内容	完成度
5	企画および仕様	20%
	課題の概要	
6	BREW 開発環境	25%
	サンプルと課題の説明 開発実習	
7	進捗状況報告	30%
	開発実習	
8	BREW 開発環境	60%
	実機での試験 開発実習	
9	BREW アプリの試験方法	70%
	検証試験 開発実習	
10	成果発表	80%

3.6 教育訓練プログラムの総合評価

この教育訓練プログラムは、総合的に以下に示すように評価できる。

- (1) 企業においてアプリケーションソフトウェアを自身で開発した経験が豊富な講師による講義と実習が行われ、受講生の理解度は、第2回を除いて3から4の間の値となり、概要理解を超えて詳細理解の高さのものであったといえる。数人の学生は理解度5を示しており、これらの学生は、活用できるレベルの高いレベルに達しているといえる。学生にとって、はじめての内容であり、はじめの数回の講義では、講義内容が具体的なイメージがわきにくかったことにより、理解度が低かったと判断できる。エミュレーションを実行し、携帯用アプリケーションプログラムの仕組みが分かりだした段階では、理解度が向上した。
- (2) 実習の時間には、プログラム開発の詳細に精通しているエンジニアのサポートを得て適切な指導がなされ、学生のソフトウェア実習におおいに役立った。
- (3) 全体として時間が不足していた。そのため、特に講義の初期には、理解度が低めになっている。特に、表3の「Cプログラミング」と「プログラム開発手順」を見ると、講義開始前よりも修了後の方が、理解度が低くなっていることが分かる。このことは、学生はCプログラミングの能力があると思っているが、実際には、アプリケーションを開発してみると、そのスキルは十分ではなかったことを自覚していることを意味する。したがって、3カ月の講義を終わってみると、学生が苦労した大きな原因はC/C++の技術レベルが十分でなく、技術レベルを実用的なアプリケーションが開発できるまで引き上げる時間が必要であったと考えられる。

以上から、この教育訓練プログラム組み込みアプリケーションソフトウェアを作成する実践的能力のための講座として、妥当なものであったと結論できる。しかし、図1に示すように、産業界の持つ最新技術の開示、技術者による講義など、実践的なIT教育を、産学協同で、継続的に実施するためには、いくつかの問題点があることも明らかになった。そこで、次章では、産学連携教育の課題を明確化し、その解決方法を示す。

4. 産学連携教育の課題

4.1 高等教育への産業界の要望

産業界は、必ずしも高いレベルのスキルを新卒人材に求めるわけではないが、日進月歩の科学技術に大学教育が追いついておらず、新卒人材は、即戦力として技術の第一線で働けるような状況ではないことが多い。その理由としては、(1) 大学における教育が知識偏重であり、実習を重視していないこと、(2) 技術は日進月歩であるため、最新の技術スキルを教育するのに適した教材が入手困難であること、(3) 実践的なスキルを持った教員が少ないこと、などがあげられる。このため、産業界では、社内教育でこれを補うのが一般的である。しかし、国内IT業界の競争の激化から、企業には、以前ほどの余裕がなくなっており、実践的なスキル教育を、高等教育機関に求めたいと考えている企業は多い。

また、産業界が求めるスキルとは、単にプログラムが書けるということではなく、プログラムの設計からユーザへ提供するまでの“一連の流れ”を理解し、品質管理の考え方ができていることを含むものである。今日では、学生がPCを持つのはあたりまえであり、プログラミングスキルを持つ新卒者は少なくないが、商用のアプリケーションを作るための一連の流れを理解していないことが問題と考えられ、このギャップを埋めるような教育を期待している⁷⁾。なお、ここでいう“一連の流れ”とは、ビジネスを明確に企画し表現する「企画書」作成、アプリケーションの動作を定義する「画面遷移図・詳細画面設計・状態遷移表」作成、試験仕様・試験手順策定、検収試験、アプリケーション検証、という一連のステップである。

4.2 高等教育の課題と解決法

上記の企業の要求に大学教育が応えるためには、教育コスト、教育方法、制度上の課題がある。ここでは、その課題と解決策をまとめる。実践的なスキル教育を継続的に実施していくうえでの阻害要因、課題を整理し、「コスト」、「教育方法」および「制度」の3つの側面から分析する。分析の対象となる課題を整理すると、以下のようにまとめられる。

解決すべき課題

a. コスト上の課題

- (1) 開発環境(ソフトウェア・試験環境など)の整備・運用にかかる費用
通信関連を含む組み込み機器の分野では、プラットフォームのバージョンアップが、かなり短い期間で実施されており、現状の安定版の規

格による教育内容は、すぐに陳腐化するものと考えられる。たとえばここで対象とする携帯電話用アプリケーションの開発では、2002年に BREW2.0, 2003年に BREW2.1, 2005年に BREW3.1と、わずか1~2年の間に互換性の少ないバージョンアップが行われている。また、携帯電話の機能も年々急増している。このため、開発環境(携帯電話, SDK)や教材(講義資料, サンプルソフトウェア)などの更新が毎年必要とされ、また、携帯電話の契約を維持するための費用負担も必要となる。このように、開発環境の整備・運用に対する負担が大きいことが、実践的な教育を実施するうえでの最大の課題であるといえる。

(2) 講師料

本教育プロジェクトで目的とする、組み込み機器のためのソフトウェア開発については、比較的低速なハードウェア上での実時間性の担保など、単なるソフトウェア開発の知識のみでなくソフトウェア開発上のノウハウに属する部分の教育が必要となる。このため、組み込み機器特有の知識が教職員に求められる。しかし、このようなノウハウ部分の知識は、それぞれの機器独特のものであり、さらに、同一の構成の機器であっても、ハードウェアごとに異なる部分も存在し、機器別、機種別のノウハウなしでは、ソフトウェアの動作自体がおぼつかない場合も考えられる。このような特化した部分の教育を、教職員のみで行うのは現実的ではなく、専門家の支援が必須となるが、時間単価が高価な第一線のソフトウェア開発技術者を講師として雇用するのは、コスト上困難である。

b. 教育方法に関する課題

(1) 学生のバラツキ

IT関連の教育を行う場合、一般に学生のレベルにバラツキが多い。特に、今回の教育プロジェクトの受講対象者は、C/C++のプログラミング経験が必要であるため、ひととりのプログラミング技術を習得した大学院レベルの学生を対象としているが、経験のレベルは学生ごとに異なる。また、携帯電話のAPIはC/C++とは大きく異なる構造を持っているため、理解は容易ではない。このため、学生のモチベーションを維持することが困難であった。また、第一線の技術者として、本教育プロジェクトの対象とするシステム自体の開発に従事した外国人の

講師を招いているが、学生の英語のレベルにもバラツキがある。

c. 制度上の課題

(1) カリキュラム上の問題

カリキュラムは、原則的には入学時に確定しているもので、これを変更するという事は難しい。このため本講義を、正式に導入するためには、新入生を対象にカリキュラムを変更し、その学生が受講する学年にならないと講義を実施できない。

(2) 講義日程の制約

講義の内容によっては、週2回程度のペースで実施した方が効果的なもの、予習復習時間をたっぷり与えて2週間に1度程度で実施したいものなどがあるが、現行の講義日程では、他の授業との重複などがあるため、フレキシブルな授業時間の設定が困難である。

(3) 実ネットワーク使用の問題

今回の講義は、試作ソフトウェアの動作確認などで、商用の実ネットワークにアクセスするケースが多々あり、そのためには試験モードの携帯端末が必要となる。今回は、実ネットワークを運用する事業者が協力して、試作ソフトウェアによるネットワークアクセスの管理が行われているが、試験モード端末や BREW 開発者だけに提供される NDA 対象となっている情報の学外流出の防止について、厳格なセキュリティ基準による管理が必要となる。

(4) 講師登用の条件

今回は、実務的に優れた第一線のソフトウェア開発技術者を講師として招いて教育プロジェクトを実施しているが、正式にソフトウェア開発技術者を講師として任用するには、論文数などの制度上の問題がある。

課題の解決法

今回の教育では、産学協同により、開発環境の整備、実ネットワーク(携帯電話)を利用した動作実験、e-learning 教材の構築などにより上記の課題に対し一定の解決が行えた。この経験をベースに残された課題への対応を検討した結果、以下の解決法が考えられる。

a. コスト上の課題

(1) 開発環境(ソフトウェア, 試験環境など)の整備・運用にかかる費用

開発環境の整備、運用にかかわるコストに関する課題を根本的に解消するためには、産業界における機器仕様の統一化、バージョンアップ

の際の、コンパチビリティの徹底した確保が必要である。このためには、ISO, IEC, ITU-T, IETF などの標準化組織と強固な連携を持ち、日本国内標準の国際標準化の積極的な推進が必要となる。このような、標準化政策に立ち入らず、この課題を解決するためには、組み込み機器に関する情報を企業側の協力を得て入手し、環境や教材を整備する必要がある。このため、大学を BREW の開発者として登録することも必要である。同時に、大学側にも専門の担当者を置き、機材やサンプルコードのアップデートを行う必要がある。

(2) 講師料

講師料の課題を一部解決するため、予習、復習用に e-learning システムを構築し、これを利用することを目指す。継続利用にあたっては、試作コードの著作権関係の問題をクリアしておくことが必要である。このような手段により、当面の講師料を低くおさえることが可能となる。

b. 教育方法に関する課題

(1) 学生のバラツキ

学生のプログラミング経験に関するバラツキは、受講経験者で成績優秀者を TA (Teaching Assistant) として任用する解決法が最善である。さらに、これら成績優秀者を企業に派遣し、実践的なノウハウをさらに習得させることが望ましい。外国人講師による教育も効果的であるが、このための英語レベルのバラツキの解消も課題となる。

c. 制度上の課題

(1) カリキュラム上の問題

既存の学部学生の、カリキュラムの全面見直しは、入学時のカリキュラムからの変更となるため現実的でない。そこで、本授業の継続については、卒業論文関連の授業として次年度カリキュラムを計画した。

(2) 実ネットワーク使用の問題

実験環境の運用方針を明確化したうえで、今後も通信事業者が協力する形で授業を継続する必要がある。このため、セキュリティ対策などを明確にし、継続して協力を得られるようにする必要がある。

(3) 講義日程の制約

本教育プロジェクトの実施により、これまであまり行われてこなかった、フレキシブルな授業時間の設定の試行導入を検討する必要がある。

(4) 講師登用の条件

ソフトウェア開発技術者の講師への任用については、当学の教授などが授業を行い、ソフトウェア開発技術者がそれを補助するというような、運用による現実的な解決を図った。本教育プロジェクトの実施により、ソフトウェア開発技術者の講師への任用についての、制度の見直しを検討する必要がある。

5. 今後の IT 教育展開への評価

5.1 大学における評価と今後の展開

- (1) 大学の問題意識や産学連携に対する姿勢の変化わが国においては輸入科学技術でしのぐことのできた時代は終焉を迎え、独創性の高い成果を示して、世界のリーダーシップをとっていくことを迫られている。本教育プロジェクトの実施の結果として、教育機関および企業が連携して教育の推進を図っていく必要性を再認識することとなった。

そのため、以下の基本方針を策定した。

- 時限付きの組織、人員を確保し、即ビジネスに結び付くような知識を持つ人材育成も視野に入れた教育を実施する。
- 基本方向として、産学連携により、わが国の人材育成に貢献できるよう、学術的な観点に加え、産業界に即貢献できる理工学、農学、医学、経済学などの人材育成を目的とする。
- 同時に、複雑な要素が絡み合う学際的な分野における先端的研究も見渡せる研究者の能力アップを図り、かつ、この分野における人材を育成する。
- この基本方針に基づき、プログラミングを主体とする技術者育成と、システム全体を見渡せる統括システム技術者を輩出できる組織の進捗管理教育を行うカリキュラムの導入を検討している

(2) 参画企業側の問題

IT 技術者は、企業にとっても重要である。IT 人材の産学一体での育成に向けた枠組みを支援する方策が、企業側にも必要である。企業から提供するものとして、教材、講師がある。これらについては、業務の一環として対応できるようにする必要がある。また、教育をうけた人材を活用する対策として、インターンシップの導入が必要であると考えている。

5.2 受講者の評価・理解度からの評価

受講者の評価や理解度については以下のとおり集約できた。これを総合すると、今回実施した教育の有効性は高く、継続に関する問題点はないといえる。

- (1) 産学連携による組み込みソフトウェア技術の習得には、携帯電話用アプリケーションソフトウェア開発のような現実に即したテーマの採用が、学生のスキル習得の有効性が高く適切なものであると評価できる。
- (2) 学生にとって身近な機器に関するテーマは、最先端の技術の習得に加え、開発管理の手法についても学生にとってインセンティブがわくもので、良い内容であったといえる。
- (3) この教育訓練プログラムの実施時より、受講した学生のソフトウェア開発の能力の向上が見られた。特に、ソフトウェア開発に必要な、計画書作成、仕様定義、テスト項目作成などの作業を実際に実施でき、有効であった。
- (4) 1人ずつでの実習では限られた実習時間内での実施が困難な課題を、プロジェクトベース学習(PBL: Project Based Learning)により、グループのメンバー間でコミュニケーションをとりつつ行った。実行計画作成や進行管理などプロジェクトマネジメントは、これまで大学のカリキュラムにはないものであったが、このプロジェクトの実施により、その基本的スキルの習得ができた。

5.3 講師からの評価

今回の教育訓練で起用した米国人のインストラクターとともに、米国のIT技術教育と今回の教育を比較した。それをまとめると、IT教育にとって重要な課題は総合的な知識を身に付けることであり、今後のIT技術教育には、以下の3点を考慮することが必要である。

- アプリケーションを記述する言語と、アセンブラレベルの言語の習得
- 情報科学以外の学問、特に人文科学系スキルの習得による、文書作成・コミュニケーション能力の習得
- 企業におけるインターンシップによるスキル向上と社会への適応

6. 産学協同IT教育への提言

IT技術教育を充実させ、人材を早期に育成するためには、産学協同して推進することが必要である。このために実施すべき施策を以下に述べる。

- (1) 企業から講師派遣をするための予算措置を行う

こと

組み込みソフトウェア技術の教育を大学で実施する際、企業から講師を派遣してもらうことが有効である。企業から大学に講師派遣する際の人件費相当の費用を大学側で予算確保することが望ましい。企業から講師を派遣する際には、企業の業務と両立させることを考慮して、3カ月間程度の短い期間とし、客員教授などの処遇をして大学教育に参加することが考えられる。

- (2) 大学の教員のスキルアップをはかるために制度(たとえばサバティカル)を導入すること
組み込みソフトウェア技術においては、大学において教員が最新技術についてスキルアップをはかるために制度(たとえばサバティカル)を導入することが望ましい。最新技術を駆使する組み込みソフトウェア技術では、マニュアルや教科書などのドキュメンテーションが整備できていないこともあるので、大学で実習を担当する教員や助手あるいはTAが、企業において技術研修を行うことができることが望ましい。
- (3) 学生が企業において、組み込みソフトウェア技術の習得を行うインターンシップを充実すること
組み込みソフトウェア開発は、アプリケーション分野によって種々異なっており、企業によってそれぞれ独自の技術を使っている。この技術を習得し、かつ企業にとってもメリットがある方策は、長期のインターンシップである。組み込みソフトウェア技術の人材を必要とする企業が、インターンシップの期間を2~3カ月として、全国の大学からインターンシップ希望者を募集し、事前に試験あるいは能力評価により選考し、選ばれた人に対して企業の持つ開発課題で適当なものを実施するようにすることが望ましい。
- (4) 組み込みソフトウェア教材と実習環境を産業界から大学に提供し、大学で通常の実験実習科目の中で実施できるようにすること
教材は最新の技術が盛り込まれている必要があり、産業界の実際の現場から、企業研修で使用するものを大学向けにアレンジして使用すると、生きた実習となり即戦力を身につけられる。
- (5) オンデマンド学習を導入すること
プログラミングをとまなう実習では、学生のレベルが個人によって大きな差異がある。学生のペースで学習が進められるように、オンデマンド学習と教室での実習をあわせて実施すること

が望ましい。そのための教材や学習環境を産学協同して開発することが望ましい。

- (6) 学生が習得するスキルを標準に照らして評価できるようにすること
公的あるいは業界標準的な検定試験あるいは資格試験に連動して、習得したスキルが就職する際の習得技能として目に見える形に結び付けることにより、プロフェッショナルとしての認識高揚と学習意欲の向上を図ることができる。

これらの施策を実現するために解決すべき今後の課題として以下の7点があげられる。

- (1) 組み込みソフトウェア分野の低位レベルのスキルについて、スキル項目と評価基準を詳細に作成し、講座の評価と個人のスキル習得に活用できるように整備する必要がある。
- (2) 組み込みソフトウェア分野のエレメンタリーレベルにおいては、大学あるいは大学院レベルでスキルの習得状況が個人に分かりやすくすることが望ましいため、個人のスキルを測定する方法を整備し確立して、利用しやすくすることが望ましい。
- (3) 組み込みソフトウェアシステム技術の実習には、「システム」開発として、企画書作成、動作定義、ソフトウェア設計、ソフトウェア作成、試験仕様作成、検収試験という一連のステップを含む必要がある。これらの過程を学生に順調に実施させるためには、学習する内容が多いため、授業実施計画を綿密にたてる必要がある。
- (4) 大学で組み込みソフトウェア技術の実習を行う場合、教育的意義が高く維持できるよう携帯電話用アプリケーションソフトウェア技術の位置づけと実習の意義を明確にすることが望ましい。
- (5) 実際をよく知っている企業の技術者が講師を務め、ソフトウェアのよく分かったエンジニアの支援を受けるようにすることが望ましいので、経費面でそのための大学における対応がとられることが望ましい。
- (6) 実習指導を円滑に行うために、学生のプログラミング技術についての能力のバラツキが少なくなるように、事前テストなどの方法により受講学生を選別して実施するようにすることが望ましい。
- (7) PBLにおける学生の修了時評価については、学生の貢献と役割を持ち点などによる加点法を採用するなど、個人の努力を適切に評価する方法をとることが考えられる。

7. おわりに

1990年代からのIT技術の進展は、New Economy⁸⁾という概念を生み出した。産業革命は、商品の種類とその生産方法を変革し、同時に、生産の場所も変えた(郊外から都市部へ)。これに対し、New Economyは、アイデアの生み出し方の変革と、アイデアを生むことの重要性を示した。このようにして発生した Weightless Economy は、商品生産のグローバル化により商品の生産の場所と方法に大きな影響を与えると同時に、Flat化と多様化と不安定化を世界的に生み出している。このような状況の中で、日本が今の国際社会での地位を維持するためには技術力の向上が必要不可欠であることはいうまでもない。少子高齢化に向かう現在、コアとなる技術を選択し集中的に開発力を強化する必要がある。組み込みソフトウェアは、今後強化が必要となるコア分野であるが、その開発には、コーディングだけでなく、企画立案から製品化までをこなし、実際的なスキルを備えた有能な人材を幅広く多数育成する必要がある。このために整備すべき現実的な課題を解決し、企業の要求と大学教育のギャップを埋めるための教育手法の開発や課題を明確化したが、1大学、1企業では対応が難しいものばかりである。今後は、このような施策への、国レベルでの支援が望ましいと考えられる。

謝辞 平成17年度経済産業省委託産学協同実践的IT教育促進事業の実施にあたり、経済産業省およびみずほ情報総研の皆様にお世話になりましたことを、ここに深謝します。また、BREWのインストラクタをしていただいた、パディ・コミュニケーション株式会社酒井紀行氏、GBW Systems株式会社 Gerrit van Wingerden氏、Joshua Done氏に感謝します。

参考文献

- 1) <http://www.keidanren.or.jp/japanese/policy/2005/039/index.html>
- 2) 長島ほか：地元企業と連携したPBL教育の実践と教育効果、工業教育、Vol.54, No.2, pp.87-91 (2006).
- 3) <http://brew.qualcomm.com/brew/ja/>
- 4) http://acpass.acpa.jp/acpass/std_capability.php
- 5) 「平成17年度経済産業省委託産学協同実践的IT教育促進事業」成果報告会資料(2006.3.10).
- 6) 茂木ほか：BREWプログラミング実践バイブル、インプレス(2004).
- 7) 南埜宜俊：魅力ある大学院教育イニシアティブ「先導的教育融合プログラム」の取り組み、創造

教育シンポジウム, 大阪 (2006.3).

8) Stiglitz, J.E.: *Economics, 3rd edition*, W W Norton & Co Inc. (2001).

(平成 18 年 5 月 22 日受付)

(平成 18 年 11 月 2 日採録)



伊藤 篤 (正会員)

1959 年生。1981 年名古屋大学工学部電子工学科卒業, 1983 年同大学院情報工学専攻修了, 同年国際電信電話株式会社に入社。1985~1999 年研究所にて仕様記述言語, 通信ソフトウェア開発環境, IN, インターネット等の研究に従事。1991~1992 年スタンフォード大学 CSLI 客員研究員。現在, 技術開発本部開発推進部所属, KDDI 研究所兼務。医療の IT 化, サイバーコート, イーラーニング, アドホックネットワーク等の研究に従事。電子情報通信学会, ACM 各会員。



渡辺 裕

1950 年生。1973 年早稲田大学理学部応用物理学科卒業。1979 年同大学院博士課程修了。工学博士。1979 年国際電信電話株式会社入社。以来, 研究所において, 国際網の計画・運用に関するトラヒック問題の研究に従事。2005 年宇都宮大学工学部教授。情報工学科に勤務。情報通信ネットワーク, 通信セキュリティー等の研究に従事。1983 年(社)電子通信学会学術奨励賞, 論文賞受賞。1986 年(財)電気通信普及財団テレコム自然科学賞受賞。1998 年科学技術庁長官賞研究功績者表彰受賞。電子情報通信学会正会員。



樽松 明 (正会員)

1961 年早稲田大学理工学部電気通信学科卒業, 同年国際電信電話株式会社入社, 研究所にてパターン認識, 音声情報処理ほかの研究に従事。1986 年 ATR 自動翻訳電話研究所社長。1993 年電気通信大学電子工学科教授, 1997 年同大学院電気通信学研究科教授, 2004 年同大学名誉教授。2004 年早稲田大学理工学総合研究所客員教授。2005 年実務能力認定機構専務理事。専門は音声情報処理, ヒューマンインタフェース。工学博士。



村上 仁己 (正会員)

1974 年国際電信電話株式会社入社以来, 研究所にてデジタル衛星通信, デジタルテレビ伝送, テレビ方式変換等の研究開発に従事する。1997 年に取締役研究所長に就任。以来, マルチメディア事業本部長, 研究開発本部長, ネットワーク技術本部長, 技術開発本部長を経て 2006 年 4 月より現職。TTC 標準化議長, ITU SG9 副議長, 映像メディア学会(旧テレビジョン学会)副会長を歴代就任。現在, 本社技術統轄部門で, コピキタス, 通放融等の技術開発および実用化に従事。



湯山 一郎 (正会員)

1946 年東京生まれ。1970 年東京大学工学部電気工学科卒業。同年 NHK に入局, 1975 年より NHK 放送技術研究所にてハイビジョンのシステム設計, 信号処理, 画像符号化の研究, EDTV, 立体テレビの研究開発に従事。立体テレビ, デジタルテレビを含む次世代テレビの研究を指導推進した。現在, 宇都宮大学工学部情報工学科教授として, マルチメディアコンテンツ, 特にパーソナルコンテンツの研究を行っている。



宮道 壽一

1971 年東京工業大学理工学部電子工学科卒業。1973 年同大学院修士課程修了。1976 年同大学院博士課程修了。工学博士。同年東京農工大学助手。現在, 宇都宮大学工学部教授。情報工学科に勤務。制御理論, システム理論, 画像処理, 画像計測, 量子計算機, 暗号理論等に興味を持つ。電子情報通信学会, 電気学会, 計測自動制御学会, 人工知能学会各会員。電気学会理事, 電子・情報・システム部門部門長。



長谷川光司

1966年生。1989年芝浦工業大学工学部通信工学科卒業。1994年電気通信大学大学院博士後期課程修了。博士(工学)。1995年電気通信大学電気通信学部助手。1996年宇都宮大学工学部助手。2001年宇都宮大学工学部助教授。情報工学科に勤務。視聴覚相互作用、音場制御、デジタル信号処理に関する研究に従事。1995年(社)日本音響学会粟屋潔学術奨励賞受賞。日本音響学会、映像情報メディア学会各正会員。



細島美智子

1978年宇都宮大学農学部農芸化学科卒業。同年宇都宮大学工学部情報工学科技官。現在、宇都宮大学工学部技術部情報処理技術班所属。読みやすいVDT文章表示についての研究に従事。照明学会正会員。



月川 淳(正会員)

1995年宇都宮大学工学部文部科学技官。並列分散処理、並列アーキテクチャ、言語処理系に関する研究に従事。現在は同大学技術部に所属。次世代マルチコアプロセッサのシミュレーションシステム環境の研究開発等を行っている。



藤井 雅弘

2003年東京理科大学大学院基礎工学研究科博士課程修了。同年同大学基礎工学部助手。2006年宇都宮大学工学部助手。現在に至る。無線通信、ユビキタスネットワークの研究に従事。博士(工学)。電子情報通信学会、情報理論とその応用学会、IEEE各会員。