2008－CE－95 (13)
2008／7／12

オブジェクト指向プログラミングの一般情報教育における意義

青木浩幸[1] 崔淑敬[2] 金顯哲[1] 李元揆[1]

[1]高麗大学 [2]ソウル市政開発研究院

オブジェクト指向技術はソフトウェア業界の常識となり，専門教育で扱われるのは普通になったが，一方の一般情報教育において，オブジェクト指向プログラミングを教えることにはどのような意味があるだろうか。高等学校でオブジェクト指向プログラミングが教育課程に取り入れられる韓国での取り組みから考察し，授業を提案する。また難しいといわれるオブジェクト指向を理解するのにはどのようなプログラミング課題が適しているか，既存のプログラミング課題の分析と併せて検討する．

Significance of Object Oriented Programming for General Informatics Education

Hiroyuki AOKI[1], Sookkyoung CHOI[2], HyeongChol KIM[1], Wongyu LEE[1]
[1] Korea University, [2] Seoul Development Institute

Object Oriented is so popular technologies for professionals that it is taught frequently for technical training. Whereas it is not clear what mean it has for general informatics education. This study has considered of Korean situation that the object oriented programming will be taken into the new curriculum, and proposes how to make the programming classes. To understand object oriented concept recognized as difficult, the subjects of the programming activity are considered from analysis of meaning of commonly used ones.

1. はじめに

韓国は2007年3月に新しい情報教科教育カリキュラムを公示した。新課程では多様な情報処理技術を身につけるようになっており，応用ソフトウェアによる情報処理に重点が移った現教育課程（2003年～）で記述が消えたプログラミングが復活することも大きな特徴である[1]。

オブジェクト指向プログラミングが高等学校の課程に入っているのも特色である。しかし，教員の間に，オブジェクト指向プログラミングが何のために，どのようにして教えるものなのかの認識は未だ不足しており，2011年の実施に向け教材研究と啓蒙が急務となっており，オブジェクト指向をなぜ一般情報教育で教えるかについての答えが求められている。

日本では2012年から（高校は2013年から）実施される次期教育課程が発表され始めている[2]。中学校の新しい学習指導要領では技術・家庭科の選択単元扱いだったプログラムによる計測・制御が必修になった。高等学校の新教育課程は未だ発表になっていないが，中学校におけるプログラミングの必修化は，現状の高等学校においても何らかの影響を及ぼすことが考えられる。日本の情報教育においてもプログラミング教育を系統的に考えていく必要があるのではないか。

本稿では，まず一般情報教育でプログラミングを教える意味について考察し，日本と韓国の現状を紹介する（2章）。その後，オブジェクト指向の導入可能性，またその意義について考察し（3章），その目的に適した題材の検討（4章），効果的に教えるための授業を提案し（5章），その授業を，大学の教養科目として試行した実践を紹介する（6章）。

2. 一般情報教育でのプログラミング

2.1. 人間が楽をするためのプログラミング

一般の人にとってプログラミングは学ばなくてはならないものではない。しかし，プログラミングをしない人とする人とで応用ソフトウェアの使い方を観察すると，しない人は習った機能しか使おうとしないのに対し，する人は自分で欲しい機能を探す傾向が見られた。これは検証していきたい問題である。

プログラミングをする人は，コンピュータは自分の仕事や問題を楽に解決するための手段という意識があるようである。一般情報教育でプログラミングを扱うにあたって，「プログラムは人間が楽をするために行う」ということを理解させることが重要と考える。

人間が楽をするためにはコンピュータに仕事を指示しなければならないが，コンピュータは人間と違って臨機応変に対応できないため，人間側が作業内容を正確に理解し，整理して指示する必要がある。結果的にプログラミングの活動は，作業を分析する力を養う効果も期待できる。授業終了後，学習者自身がプログラムをすることはなくても，仕事の本質的な理解や，自動化に対する意欲を持つことが，情報活用能力につながると考えられる。

コンピュータが高機能化し，いろいろなことを自動的にやってくれるようになって，楽ができるようになってはいるが，その「楽になる」とは本質的に異なることに注意したい。コンピュータの高機能化による「楽」は人間を堕落させるが，プログラミングによる「楽」は人間を利口にさせる。

楽をすること実感するためには，人間がした作業より何倍もの結果がプログラムにより得られることが有効である。そのためには題材の選択が重要である。適切でない例として，C 言語の入門で有名な「Hello World」[3]の題材がある。コンピュータに表示させるより人間が自分で打った方が楽なため，C 言語を学ぶことを目的としない者には意義が感じられないと言われる。アルゴリズムの尺度に計算量があるように，プログラムによってどれだけ人間の作業を省略できたかを示す「作業量」という概念を課題プログラムの評価の尺度に提案したい。

2.2. 一般プログラミング教育の現状

(1) 日本の状況

中学校では，技術・家庭科の「プログラムと計測・制御」の単元で順次・選択・反復の基本 3 構造を学ぶ。数あてゲームのような題材で値の入出力を繰り返したり，センサを常時監視したりする実習例が教科書にあり，人間の替わりに人間の相手をしたり，監視をしたりしてくれる点で「プログラミングによって楽をする」ことを学べる。

高等学校では教科書では，情報 B の(2)イにおけるアルゴリズムの理解と(3)アのモデル化とシミュレーションでプログラミングが扱われている。しかし，問題の定式化，正しさの検証という段階を明確にしていないという問題が指摘されている[4]。つまり，与えられたプログラムを打ち込んでそれを実行してみるという体験にしかならない可能性がある。

これは，学習指導要領の縛りのため，プログラミング言語を教えることに遠慮があり，自分でプログラムを作ることができる知識を与えていないことが大きな要因である。仕事の内容を自分で考えてコンピュータに指示するのでなければ，本質的な「楽をする」理解にならない。打ち込むだけの実習が本当に行われるならプログラミングによる教育の評価を下げるだけである。

(2) 韓国の状況

新しい教育課程を目前とした韓国では教科書作りが進行中で，プログラミングについても検討が進められている。情報教育に関する教科は必修にはなっていないが，どの年齢でどの程度のプログラミングができるかという発達段階や，系統性について参考にすることができる。

小学校では週 2 時間の学校裁量時間もしくは 5，6 年生に設定された実科の中でコンピュータを用いた活動が設定されている。研究としては MIT メディアラボが開発した Scratch[1]の発表数が多く，教科書への採用が有力視されている。こ

[1] http://scratch.mit.edu/

れはアニメーション製作やゲームに適しており，「ぶつかったら何をするか」,「キーが押されたら何をするか」といった，イベント駆動的なプログラミングである。配列がなく，データ処理には向いていないので「人間が楽をする」視点はない。

日本では高等学校で扱われていた整列や探索アルゴリズムは中学校に配置されている。その分，高等学校では構造的プログラミング，オブジェクト指向プログラミングが内容となっており，プログラミング利用による学習にとどまらず，ソフトウェア工学のような情報科学の基礎を学ばせようという意図が垣間見える。

高等学校にオブジェクト指向プログラミングが入っているが，何のためにオブジェクト指向プログラミングを教えるのかという認識は未だ広まっていない。そのような背景で，我々はオブジェクト指向を教える意味について研究を進めている。

3. オブジェクト指向プログラミングの導入

3.1. オブジェクト指向は無理か

「オブジェクト指向」はJavaの躍進や，幅広くソフトウェア開発に応用されるに伴い，社会的にも有名な言葉になった。今どきプログラミングを学ぶならオブジェクト指向言語でなければ，という声さえ聞かれる。

しかし一方でオブジェクト指向は難しいと言われる。本屋には，入門者にオブジェクト指向を分かりやすく説明するための本がたくさん並んでいる。

難しいのは，プログラミングは楽をするために行うという視点が抜けているからである。クラスの抽象的な説明を聞いて，多くのおまじないを打ち込んで，コンピュータにさせることができた仕事が単に「Hello World」の表示では[5]，プログラミングは難解であって特別な人がやるものと思われても不思議ではない。

オブジェクト指向の意義が活かされていないため，学習者にとって多くのことが無駄な努力になっている。オブジェクト指向プログラミングが効果を発揮するのは大規模ソフトウェア開発においてであり，プログラミングの入門コースには適さないという指摘もある[6]。初心者は通常の手続き型プログラミングによって，プログラミングと計算機の基本構造を理解した上で，抽象度の高いオブジェクト指向プログラミングに進むべきで，プログラミング教育修了者に対するソフトウェア工学コースで導入することが望ましいとしている。

これらのことから言えることは，オブジェクト指向言語を学ぶことを目的にしてはいけないということである。あくまで，楽をするためのプログラミングを理解するのに，オブジェクト指向が役に立つという姿勢が大切である。

3.2. 導入する意義

オブジェクト指向は情報化社会の発展によるソフトウェア危機に対抗するために生まれた，複雑化するプログラミングをやさしくするものである[7]。では初心者の作るプログラムは簡単であるため，オブジェクト指向のような仕組みは必要ないかといえば，必ずしもそうとは言えない。それは，現代のコンピュータは初心者にとって十分複雑なものになってきているからである。

どうしてオブジェクト指向によりプログラミングがやさしくなるのか，その要因を以下のように整理する。

(1) かくす働き
　意識する必要のない部分を隠ぺいすることで，主題に集中できるようになる。部分部分の独立性を高め，再利用性を高くする。

(2) まとめる働き
　対象に対する情報をまとめ，役割分担をする（→オブジェクトの定義）。整理整頓がされ，プログラミングをする者にとって名前付けや検索がしやすくなる。

(3) たくさん作る働き
　オブジェクトが定義されると，その定義に従って同種のオブジェクトを大量に生成できるようになる。

(4) 枠組みの働き
　コンピュータが持つ膨大な機能を整理して

与えてくれる。また，すべてを作るのではなく，関心のあるところだけを作れるようにしてくれる。プログラムの作り方のパターンを与えてくれる。

(1)のかくす働きの特徴を利用して，工業高校の授業で，Java の機能を教材用に再構成した実践がある[8]。また，(3)の特徴はディジタル情報の複製が容易という性格と，「楽をするためのプログラミング」という原理の理解に効果的である。

(2)でオブジェクトを定義することは，対象についてどのような情報や動作があるか，分析と設計を行うことである。これは高等学校の教科情報の学習内容「モデル化とシミュレーション」と一致する。

3.3. 残された問題点

オブジェクト指向言語を使うことがオブジェクト指向プログラミングでない。オブジェクト指向言語である JavaScript が使われている高校教科情報のプログラム例を見ても，3.2. で挙げたオブジェクト指向のまとめる働きや，たくさん作る働きなどの特徴がほとんど活かされていない。それはサンプルが悪いのではなく，単にオブジェクト指向の考え方は必要ないからである。

現実世界の例えはオブジェクト指向の特徴としてよく取り上げられるが，実際は抽象的な話は現実世界を必ずしも反映しておらず，プログラミングの対象が絞り込めていない初心者にとって，混乱させると指摘されている[7]。適切な題材を選ぶ必要がある。

プログラミング言語についても適切なものを使う必要がある。初心者用のオブジェクト指向言語として「ドリトル」がある。LOGO の「マインドストーム」は CAI により「コンピュータが子どもをプログラミングする」のではなく，プログラミングによって「子どもたちがコンピュータに教えよう」という，教育におけるコンピュータの考え方に一石を投じた。ドリトルはそれを引き継ぎ，「子どもたちがオブジェクトに言葉を教え，オブジェクトを作る」という「オブジェクトストーム」という考え方を提案している[9]。

小学生から大学生にまで実践例が紹介されていて，その学習の易しさは確認されている。英語中心のプログラミング言語が多い中で，日本語や韓国語といった，学習者に理解しやすい言葉が使えることがプログラミングの敷居を低くしている。

特にオブジェクト指向の学習に向いているのは「散らかしっ放しでいいこと」である。スクイーク eToy や Scratch という，オブジェクト指向といわれる教育用言語があるが，これらは一回一回の実行が継続され，リセットされずに，オブジェクトを後始末する機構がない。そのため「楽をするプログラミング」のためにオブジェクトを大量生成する学習には用いることができなかった。

4. プログラミングの題材

日本でも高校の教科書で取り上げられているアルゴリズムを扱う内容は，プログラミングを道具として使うという位置づけである。データを対象に処理できるという点では適しているが，整列や探索アルゴリズムを学習者が自分で考えだすには発想の広がりに限界があること，その仕事に意義を見出しにくいことが難しいところである。

ゲームは小学校対象のプログラミング活動でよく用いられる題材である[10]。子どもが自由に発想でき，モチベーションが高くなるという点で，アルゴリズムやシミュレーションとは対照的である。ただし，自由であるだけ，教育の場で何かを教えることを考えた場合，その目標が定まりにくい。小学生では体験して楽しさが分かることが重要であるが，年齢が高くなるにつれ，それだけでは不十分になってくる。

人間が楽をするプログラミングのための，オブジェクト指向に向いた題材こそ逆に教材として利用できるのではないかと考えられる。

オブジェクト指向言語の歴史は 1967 年に開発された Simula67 に遡る。Simula67 の目的はシミュレーションを作ることであった。オブジェクト指向で楽をすることを学ぶためには，オブジェクトを大量発生させるような題材が適している。原子核の連鎖反応のような題材が挙げられる。

オブジェクト指向の「まとめる働き」などは，シミュレーションのモデル化と密接でありよい題材となる可能性がある。しかし，シミュレーション対象の背景知識が求められ，自分の専門領域でなければ学習者自身がモデル化を行うのは困難であるという問題があった。

シミュレーションはただ計算をしただけでは意味がないことにも注意しなければならない。学問的に正しいとされるモデルを与えてシミュレーションを行うことがあるが，正しいものを与えて正しい結果が出るのは当たり前である。

ひとつの使い方は，正しいか分からないアイディアを試してみて，現実とマッチしていることを検証して，そのアイディアの正しさを図ることである。もう一つの使い方は，正しい結果を利用するもので，実際の世の中を疑似的に体験することである。訓練や試験，ゲームとして娯楽に用いられることもある。

ゲームは同種のキャラクターが多数登場し，オブジェクト指向に適した題材である。また，学習者自身がモデルを作りやすく，シミュレーションの応用場面でもある。専門教育ではその効果を認め，実践例が見られるが [11]，一般教育でもその効果を期待することができるだろう。

5. 授業構成とその意図

5.1. 学習計画

学習者の発想をプログラムに実現させることが，コンピュータに対する主体的な態度を育てることになると考え，作品製作を最終目標とした。作品のテーマは大量なオブジェクトを用いたゲームである。

学習者自身がプログラムを作れるように，例題によるプログラムを作る手順の概観，課題による技術の習得，作品の製作という段階を踏む。

5.2. コンピュータサブセットモデル

オブジェクトの説明にあたって，現実世界の例えを使うのは止め，これからプログラミングで実際に使うオブジェクトを使って説明した。その方が実際的メリットを理解しやすいと考えたからである。

オブジェクト指向の「かくす働き」，役割分担により，プログラミング言語で提供されているオブジェクトは，コンピュータの機能を一部切り出した小さなコンピュータと考えることができる。

膨大な機能をもったコンピュータに直接プログラミングして指示を出すのは難しいが，「タートルオブジェクト」のような機能（絵を描く）に限定されたコンピュータであれば，その機能を把握するのは容易であり心理的負荷を小さくすることができる。図 1 はそのことを説明した図である。

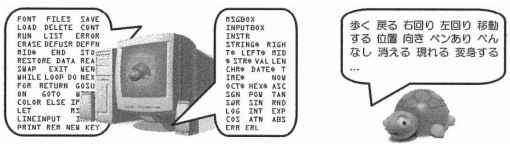

図 1 オブジェクトとは何か

このような小さなコンピュータ「オブジェクト」を組み合わせたり，機能追加をしたりして好みの「システム」を作っていく。個々のオブジェクトに指示を出す方法として，プログラミング言語の文法を学ぶ。

5.3. シミュレーション作成の例題

モデル化の手順を学ぶためにシミュレーションプログラムを例題とする。

プログラムの経験のないものには，先が見えないことが理解を妨げるので，まず図 2 のような打上花火の完成イメージを与える。

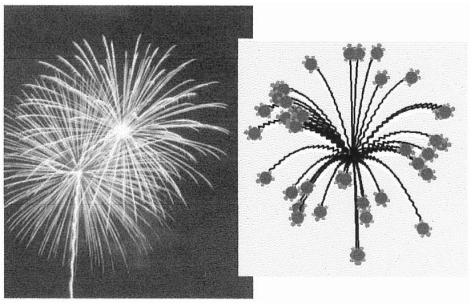

図 2 実物とシミュレーション完成イメージ

物理学的問題であるが，シミュレーションはモ

デル化の結果を検証するために行うもので，「正しい」とされる物理法則を使って答えを導き出してしまっては，正しくて当然であり，何を検証しているのか分からなくなる。また人間が頭を使って計算式を導いてしまっては，コンピュータを使って楽をするという意図にも反する。

単純な運動モデルを幾つかコンピュータに与えコンピュータに計算させることで，その結果を見てモデル化の正しさを検証することにする。与える運動モデルは，爆発によって，「星」(火の玉)は四方八方に飛び散るということと，「星」は重力によって落下するという 2 つの要素とする。それぞれの要素は直観的で，物理的背景知識がなくても理解できるものである。また，落下の方法を学ぶのは，ゲームに応用ができることを視野に入れてのことである。

この例題を通してプログラミングの過程を学ぶ。

1. オブジェクトの姿を設定
2. オブジェクトに情報を記憶させる
3. 新しい命令（動作）を教える
4. 時間ごとの変化のさせ方（タイマー）
5. 自律行動をさせる
6. オブジェクトの大量発生

1.-3. は構成する「星」についてどのような情報や動作が必要かを分析し，オブジェクトとして定義することである。4. ではタイマーオブジェクトの利用によって定期的に命令を発生し，「星」オブジェクトが動き出させる。この時点で一度プログラムを実行させ「星」が画面上で設定通りに動くことを確認する。5. はそのタイマーを「星」オブジェクトに搭載し，他から命令されるのではなく，「星」自身が自律行動できるようにする。これはゲームを作る上でも有益な技術である。

その星をタイマーによる繰り返しで多数発生させることで花火が完成する。図 3 が完成したプログラムである。プログラミングによって作業量より多くの成果が得られる実感を得られることがポイントである。

```
星＝タートル！作る
  〝ayumiaka.gif〟 変身する。
  星：落下速度＝10。
  星：飛行＝「！10 歩く。」。
  星：落下＝「
    ！0 (落下速度) 移動する。
    落下速度＝落下速度-1。
  」。
  星：スタート＝「
    爆発動作＝タイマー！作る
      0.05 間隔 1 秒 時間「
        ！飛行。！落下。」実行
      1 回数「！消える」実行。
  」。
  星発生＝タイマー！作る
    0.01 間隔 40 回数「
      (星！作る)！(乱数(360)) 向き
      スタート。
    」実行
  1 回数「星！消える」実行。
```

（星のモデルを定義（外観・情報・動作）／星のライフサイクル／星を大量発生（タイマを利用した繰り返し））

図 3 例題課題のプログラム

5.4. 課題による技術習得

ゲームのプログラムを作るには多少の技術が必要である。作品製作に有効な技術を一つ一つ紹介し，学習者自身が課題に挑戦しながら身につける。具体的にはオブジェクトの整列や時間経過の処理，衝突の処理，インタラクティブな操作である。

反復処理はタイマーによる方法に統一した。タイマーオブジェクトが命令するというモデルが理解しやすいのと，プログラムの動作を詳しく観察したい時に，注目したい部分のみをゆっくりにできる利点がある。また，タイマーに名前を付けることにより，後から動作を止められるようにするとともに，プログラムの部分部分に説明を加え見通しを良くすることを期待している。

5.5. 作品製作－ゲーム製作

授業の途中でお互いの作品を紹介する機会を設け，活発にアイディアが出せるように支援を行う。その際，日本語プログラムの読みやすさを利用して，プログラムのコードを読ませ，どのような仕組みで実現されているのかの工夫を交換することも重要である。

また，ドリトルはイベント処理と多態性を利用されるので大抵の場合で条件分岐を使う必要がないが，ゲームの勝敗にかかわる判断として条件分岐を導入する。

6. 授業実践

提案する授業について，韓国高麗大学の一般教養の授業で実践を行った。受講した学生数は 68 名で，様々な学部から集まっており，大部分がプログラミングの経験のない 1，2 年生である。授業時間は 90 分が 4 回，1〜3 人のグループで取り組み，課題を宿題として課した。図 4 は学生の作品例である。大量のオブジェクトを発生させ，シミュレーションを応用した作品が数多く見られた。

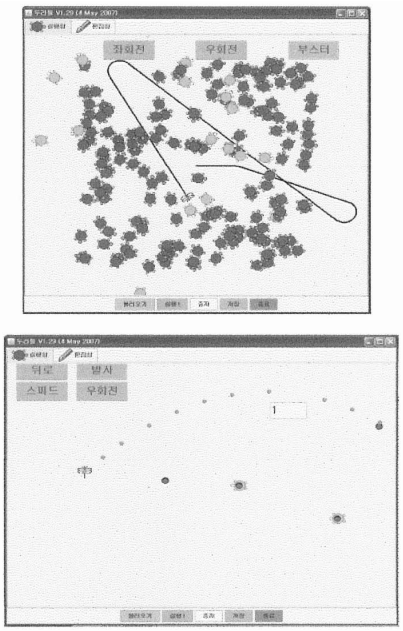

図 4 学生の作品例

授業終了時のアンケートではプログラミングを経験したことに対して好意的な反応が得られた。プログラミングへの関心が上がった学生の比率は 59%で，下がった学生は 3%であった。「課題の構成が緩やかなのでいろいろな考えを出すことができた」（訳）と，学習者の発想を重視した授業の意図を肯定する記述も見られた。一方プログラミング言語の解説不足への指摘も散見され，指導資料の問題や 6 時間という指導時間では難しいことが分かった。

7. おわりに

プログラミング言語を習得するのが目的ではない一般情報教育だからこそ，コンピュータは便利なものであるという好意的な態度を育てたい。そのためには，授業の目的を踏まえた上で，学習者自身が作りたいものを作らせる必要がある。

今回の提案のアプローチは，プログラミングの「人間が楽をするために行う」という意義を，オブジェクト指向プログラミングの特性を利用して学ばせようというものであった。

韓国の新教育課程に「オブジェクト指向プログラミング」という項目があるが，流行り的な「オブジェクト指向」という言葉に踊らされ，形式やおまじないを操ることを教えるのではなく，教育者はそこから何を教えられるかということに集中するべきである。授業を組み立てるにあたっては適材適所のプログラミング題材の検討や，様々な工夫を蓄積していくべきだと考える。本研究の取り組みがそのような議論の一助となることを期待するものである。

参考文献

1) 李元揆，変わりつつある情報教育，海外の情報教育の動向，情報処理，Vol. 48， No. 11，pp.1207-1212，2007.
2) 文部科学省，新しい学習指導要領「生きる力」，http://www.mext.go.jp/a_menu/shotou/new-cs/index.htm，2008.
3) B. W. Kernighan, D. M. Ritchie, *The C Programming Language*, 2nd ed., Prentice Hall, 1988.
4) 長慎也，兼宗進，並木美太郎，西田知博，小井土政範，「情報 B」の教科書比較ー「手順的な情報処理」の観点から，情報処理学会研究報告，2006-CE-85(8)，pp.53-59，2006.
5) 神林靖，「オブジェクト指向プログラミング教育への疑問ー初等プログラミング教育における方法論を考えるー」，日本工学教育協会平成 15 年度工学・工業教育研究講演会論文集，pp.9-10，2003.
6) Ralph Westfall, Hello, world considered harmful, *Communications of the ACM*, vol.44, No.10, 2001.

7) 平澤章:“オブジェクト指向でなぜ作るのか 知っておきたいプログラミング，UML，設計の基礎知識”，pp.82-94，日経 BP 社(2004).
8) 山本哲也，前川　仁:“工業高校情報技術科における教材クラスを利用したプログラミング学習”，第 4 回情報科学フォーラム FIT2005 講演論文集，第 3 分冊，K-007，2005.
9) 中谷多哉子，兼宗進，御手洗理英，福井眞吾，久野靖，オブジェクトストーム:オブジェクト指向言語による初中等プログラミング教育の提案，情報処理学会論文誌，Vol.43，No.6，pp.1610-1624，2002.
10) 佐藤和浩，紅林秀治，青木浩幸，西ヶ谷浩史，井戸坂幸男，鎌田敏之，原久太郎，久野靖，兼宗進，IT クラフトマンシッププロジェクト～小中学生によるドリトルプログラミング～，情報処理学会研究報告，2006-CE-83(23)，pp.165-172，2006.
11) Woei-Kae Chen, Yu Chin Cheng, Teaching Object-Oriented Programming Laboratory With Computer Game Programming, *IEEE Transactions on Education*, Vol. 50, No. 3, 2007.