

Technology Enhanced Natural Learning

Pavel Boytchev, PhD

CIST, Sofia University

Sofia, Bulgaria

pavel2004@elica.net

Abstract

Technology Enhanced Learning is one of the most discussed achievements in this century. This paper discusses the problematic issue whether the use of technology is advantage or disadvantage. What factors can be controlled in order to increase the positive influence of technology to the learning process. As a demonstration a new approach to Object-oriented programming is being briefly discussed - the so-called Natural OOP, which is supposed to bring OOP to a more understandable level. This approach illustrates how complex technology in software can be represented in a way that enhances learning.

Keywords

Learning, technology, natural learning, Logo, 3D modelling, animation, Elica

1. Introduction

While thinking about *Technology Enhanced Learning* it is important to determine the exact focus. Is it *technology*? Or is it *enhanced*? Or maybe the focus is on *learning*? It appears that whatever word is assumed to be the main in this phrase, it will not represent the whole spectrum of activities and ideas covered by the Technology Enhanced Learning.

If we go back in the history we could find the most influential technology invention. It was so important, that it is still used nowadays - in schools, at homes, in libraries. This invention is the *book*. From technological point of view it is a perfect tool for learning. It has been used for centuries, and it will be used in the future. What factors do make the book so successful? Can we apply these factors to modern technology enhancements like computers and computer science?

First of all the book has a clear interface. It is not cluttered with fancy gadgets. Another major factor is that the book is intuitive. People learn to use a book since they are a few years old. The learning curve is almost a flat line. Finally, the third factor is that this is a transparent technology. All people use the book without thinking about the book itself. They don't hesitate whether they flip the page correctly even if they get a book in a foreign language that they cannot read.

If we consider the book as an ultimate example for a successful tool for technology enhanced learning, can we compete the book by using modern tools? And should we do it? Is an electronic version of a book a better tool for learning or not? All these are

questions without clear answers. Using technology can enhance learning, but it can also render it more difficult.

2. Technology, black holes and natural learning

There is no question that technology gives us a lot of advantages not only in education, but also in our everyday life. The main properties that contribute to technology advantages are:

- Technology is *attractive*. We cannot think about using technology if it is not attractive. Attractiveness could be achieved in many ways - functional, visual, ideological. Whatever way is used, technologies which are more attractive will be more used by people.
- Technology is *available*. A nice technological improvement is not valuable to the society unless it is available. This does not mean technology should be free, although many people would be more than happy. It means that technology is not locked behind the doors of research institutions and science labs.
- Technology is *addictive*. This is one of the most ambiguous benefits of technology. The addictiveness manifests itself by the choice of the people. It also reflects to the efforts they spend in acquiring, implementing and improving this technology.

Like all things, technology cannot be only good. There are some disadvantages, which are quite obvious. If technology is overused, underused or misused, its main advantages become its main disadvantages. Attractiveness, availability and addictiveness can turn down the positive use of technology.

From the perspective of human-technology interaction, technology can be treated as a black hole. If users get too close, they could be gulped down. Getting out is difficult and sometimes impossible. Some cases of technology addictions can be classified as illnesses of the modern age (e.g. addiction to Internet or video games).

If users are too far from technology, they may never experience its benefits. This is a typical example of underused technology.

The secret of proper use of technology is in the right balance. The gravitational energy of a black hole can be used to accelerate further in the space. In a similar manner technology can be used to boost learning.

To see how technology can enhance learning, it is need to distinguish two styles of the learning process - *forced* and *natural learning*. We talk about forced learning when the focus is on the learning itself. A typical example is learning a lesson by reading it from a textbook, classes in school, etc. Focus learning is every learning process during which the learner is consciously engaged with the study. Usually all additional activities are suspended. Forced learning is generally used in school years - from primary school to university.

The opposite style is the natural learning. It happens before primary school and after university. The main difference is that the learner is not focused on the learning, but on something else. The process of learning is maintained on a by-the-way basis. Early applications of this style are children's games.

The main focus during the natural learning is a goal not directly related to what is actually learned. The benefits of this style are that learners do not feel forced to learn and the process is much less boring.

To have a successful learning many educators apply various methods for naturalizing the learning by changing the focus. Examples of such naturalizations are the introduction of edutainment and games. Another possible approach is to relate complex terminology (which often accompanies the forced learning) with a terminology of a natural process.

3. Natural object-oriented programming

Object-Oriented Programming (OOP) is a collection of concepts, tools and styles that are supposed to help programmers build more complex programs which are easier to support (Digitool, Inc, 1999). Although there are many benefits in using OOP, there are cases when those benefits vanish. A typical example for the failure of OOP is its implementation. As a result, unlike professional programmer's software, amateurs are frightened when they have to learn and apply OOP.

A possible reason for this might be the fact that OOP is often introduced at the cost of changing the syntax of a language, or at least, adding numerous new commands and functions that *must* be used if one wants to use OOP. The situation is even worse, because beginners are flooded with a list of new words like *classes*, *objects*, *instances*, *methods*, *fields*, *inheritance*, etc. The effective study of OOP requires a lot of forced learning.

The Natural Object-Oriented Programming (NOOP), on the other hand, has almost the same functionality and applicability as OOP, but it is based on and explained by entirely natural terminology. The fact that all entities and processes in NOOP can be demonstrated by entities and processed in evolution makes it a good candidate for natural learning.

To evaluate whether NOOP is applicable in practical programming, it has been implemented in Elica - Educational Logo Interface for Creative Activities (Elica, 2005). Elica is a modern dialect of the Logo programming language (Boychev P, 2005b) that supports NOOP and 3D graphics and animation. So, it utilizes both ways of naturalization: simplification of terminology (the evolution metaphor in Elica NOOP) and changing focus (i.e. 3D graphics).

The big advantage of NOOP is its evolution metaphor and the lack of significant changes in the Logo language. The relation to evolution is based on the idea that all entities and processes in OOP can be expressed as entities and process in the evolution. OOP terminology like *classes*, *objects*, *fields*, *methods* and *inheritance* correspond to *species*, *creatures*, *properties*, *behaviour* and *inheritance* in NOOP. However NOOP appears to be more versatile, because it gives a meaning to *cloning*, *mutation*, *pregnancy*, *birth*, *incarnation*, *adoption*, *death*, *murder*, and much more.

Let's take one example - birth. In NOOP it can be implemented in several ways, each of which corresponds to a different natural process.

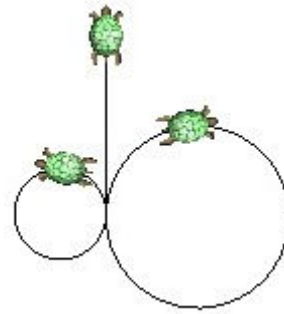
- *Cloning*. In programming this is creating by assignment. In evolution - the creation of exact copies is the way of reproduction of primitive creature. The following program makes two clones of the default turtle. All three turtles are identical but independent. Each of them in then controlled to draw a circle or a line. The default

turtle goes forward, the two clones make circles (see the program and the snapshot below):

```
run "turtle
use "turtle3d

make "a :defaultturtle
make "b :a

repeat 55
[
  suspendpaint
  fd 1
  ask "a [lt 8 fd 2]
  ask "b [rt 8 fd 4]
  resumepaint
]
```



- *Playing God.* In programming this is creating an object field by field. In NOOP this method is like playing God. The programmer defines the properties and the behaviour of a creature one at a time. The following example demonstrates the creation of two points: A(10,20) and B(0,30). Note that no class or constructor is used, objects are defined explicitly:

```
make "a.x 10 ; point A(10,20)
make "a.y 20
make "b.x 0 ; point B(0,30)
make "b.y 30
```

- *Mutation.* In programming this is creating by modification. The evolution terminology is mutation - an existing object is modified. The next example is based on cloning and mutation. There is one default turtle, which goes forward, and two clones that make circles. The mutation here is that the clones are "dressed" differently. Except for their appearance, they are identical to the default unmodified turtle.

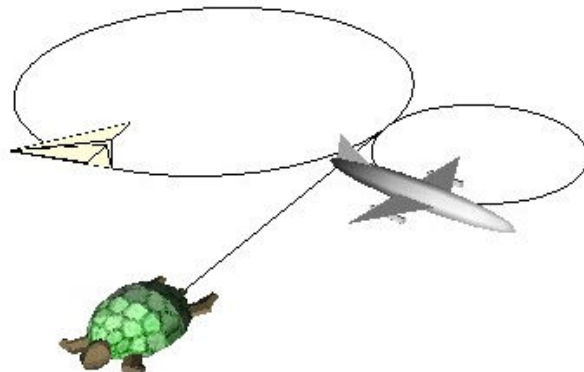
```
run "turtle
use "turtle3d

make "a :defaultturtle
make "b :a

demo 200

a.use "airplane
b.use "paperplane

repeat 55
[
  suspendpaint
  fd 1
  ask "a [lt 8 fd 2]
  ask "b [rt 8 fd 4]
  resumepaint
]
```



- Natural birth. The forth method for creating objects (i.e. birth) is the natural birth. In programming this is creating object instances from object classes. In evolution this method corresponds to the birth of a new creature from specific specie that determines its features. In the following example there is a class HUMAN with three initialization parameters - name, age and height. When this class is activated to give birth of an object, it checks the age. If it is greater than 80, then the newborn object is given a new property - glasses. The behaviour of all created HUMAN objects is that they can 'say' their name. The example also creates two objects, Karina and Tom.

```

to human :name :age :height
  if :age>80 [make local "glasses "true]
  to hello
    print "Hello "my "name "is :name
  end
end

make "a human "Karina 50 1.70
make "b human "Tom 85 1.85

```

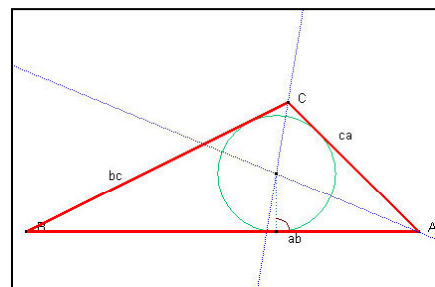
The natural thing in this example is not only in respect to the evolution. The definition of the HUMAN class is actually a definition of a procedure. HUMAN becomes a class when it is used as a function. This is a demonstration of how NOOP is introduced without increasing the complexity of the programming language. The basic syntax constructions in the language can be used to express the more complex OOP ideas.

4. NOOP Applications

The theoretical foundation of the Natural Object-Oriented Programming is a nice area of research, but the ultimate goal is to apply this into practical programming. All examples shown so far were pretty simple. The benefits of NOOP are really valid only if they persist for much larger applications.

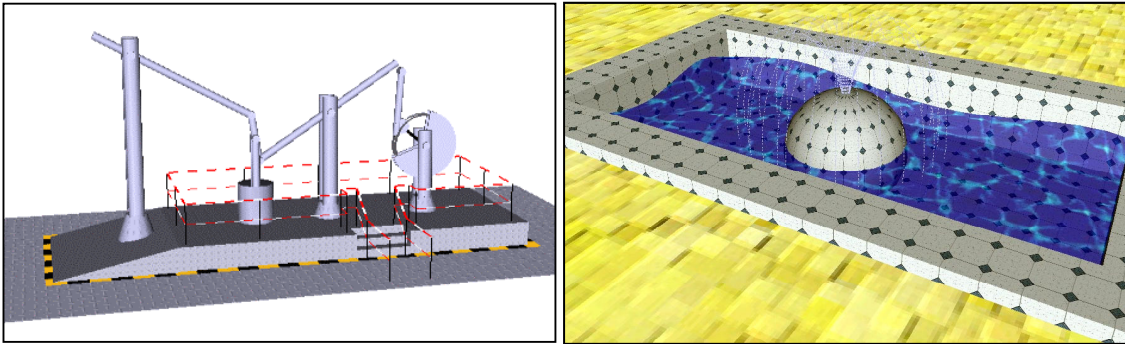
- *Geomland*

Geomland is a geometry-oriented Elica Logo library that enhances Elica core Logo. It helps students and teachers to explore various plane geometry problems by programming constructions. The picture on the right shows a red triangle with inscribed circle. The user can reshape the triangle with the mouse and the rest of the construction is automatically adjusted following the initial geometrical relations.



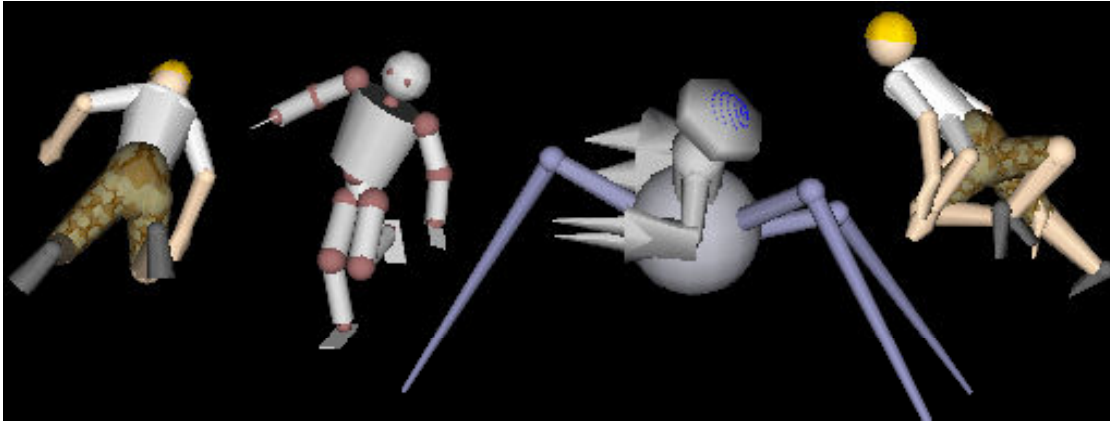
- *Simulation and modelling*

The strength of Elica is mainly in the area of modelling and simulation (Boychev P, 2005a). The examples below show a complex steam machine and a water pool (Teining V, 2005; Virtual Terrain Project, 2005). The machine and the water are animated in real-time. All graphical objects are based on NOOP.



- *3D body models*

NOOP technology is suitable for building 3D models of bodies and for animating them in real time (Boychev P, 2003a). The first two images in the snapshot below are of human and robotic bodies. The benefit of using NOOP is that it is easy to define any form of bodies. The attractiveness and the natural expression of complex bodies are a major contribution to the technology enhanced learning based on Elica.



5. Final words

Technology Enhanced Learning is a basic learning stream in the century. The application of technology has revealed that a very precise balance is needed in order to achieve good results. The failure of balancing technology may lead to negative results.

Technology Enhanced Learning is not a standalone process. It requires changes in many concomitant processes to make it a natural learning. The NOOP in Elica is just a piece of the whole picture. It is a small demonstration how natural learning and natural thinking can be applied to domains where forced learning has been dominant for years.

Although the huge amount of innovations in technology, the book still remains the most effective educational tool. The best role for computers is not to take the place of the book, but to provide an additional and complementary enhancement to the learning process.

References

Elica (2005), *Elica Home Page*, <<http://www.elica.net>>

Boychev P (2005a), *Using Logo to Model and Animate*, 10th European Logo Conference EuroLogo'05, Poland, 2005

Boychev P (2005b), *Logo Tree Project*,

<<http://www.elica.net/download/papers/LogoTreeProject.pdf>>

Teining V (2005), *Mathematical/Numerical Modelling of Water Waves*,

<http://www.mek.dtu.dk/English/Research/Feature_Articles/Mathematical_Numerical%20Modelling_of_Water_Waves.aspx>

Virtual Terrain Project (2005), *Water Rendering and Simulation*,

<<http://www.vterrain.org/Water/>>

Boychev P (2003a), *Student-oriented Software for Application of Geometry in*

3Dmodelling and Animation, Abstracts of International Congress of Mathematical

Society of South Eastern Europe MASSEE'2003, Bulgaria, 2003

Boychev P (2003b), *Turtle Metamorphoses (From "FD 1" To 3D Animated*

Characters), 9th European Logo Conference EuroLogo'03, Portugal, 2003

(all links last visited in December 2005)