LEGO Approach in Teaching System Dynamics

Dong-Hwan Kim

(Electronics & Telecommunications Research Institute, sddhkim@soback.kornet.nm.kr)

Back to the LEGO

In the childhood, we are used to constructing a tank, an airplane, and a house with the help of LEGO. LEGO is composed of many building blocks. We can construct a house by connecting blocks. We play LEGO with two steps: take an appropriate block and connect it into another blocks chosen before. However, within the stock-flow modeling approach, one follows three steps: create, connect, and define a variable.

Among the three steps, the last step of defining a variable equation seems to be most cumbersome. As far as the variable definition is not essential to the systems thinking but a main obstacle to learning system dynamics, it seems desirable to make the burden of students light. If we eliminate the last step of system dynamics modeling, one can go back to the childhood age of playing LEGO. One can play without comprehending the task of formulating equations. LEGO provides a fascinating analogy with which children can learn and play simultaneously. In this paper I propose a LEGO approach as a tool for allowing children to model their world.

The LEGO approach of this paper is different from Molecules suggested by Eberlein and Hines. While LEGO approach deals with single typical variables and uses a graphic icon for representing them, Molecules approach is concerned with a set of stock-flow variables. While generic models and Molecules can be used by system dynamicists, the LEGO approach proposed in this paper may be appropriate to the novices.

LEGO approach for system dynamics modeling

Most students have difficulties in formulating mathematical equations describing a concrete system. A complex equation is like a hammer. If we misuse our hammer, it will hurt our finger rather than a nail. If you are not an expert in using a hammer, you should use a nailing-machine rather than a hammer. LEGO approach can provide a nailing-machine for teaching system dynamics modeling to the children. A key point for implementing LEGO approach in system dynamics modeling is to eliminate a step for defining equations. Figure 1 compares LEGO modeling approach with traditional one. Figure 1 shows how to model a prey-predator system. In the LEGO modeling approach, I provided icons for rate variables including prey-change and predator-change. Prototype equations are embedded in these icons. As soon as one connects another variable to these icons, a simple window pops up to ask what variable in the prototype equation matches to the newly connected variable.

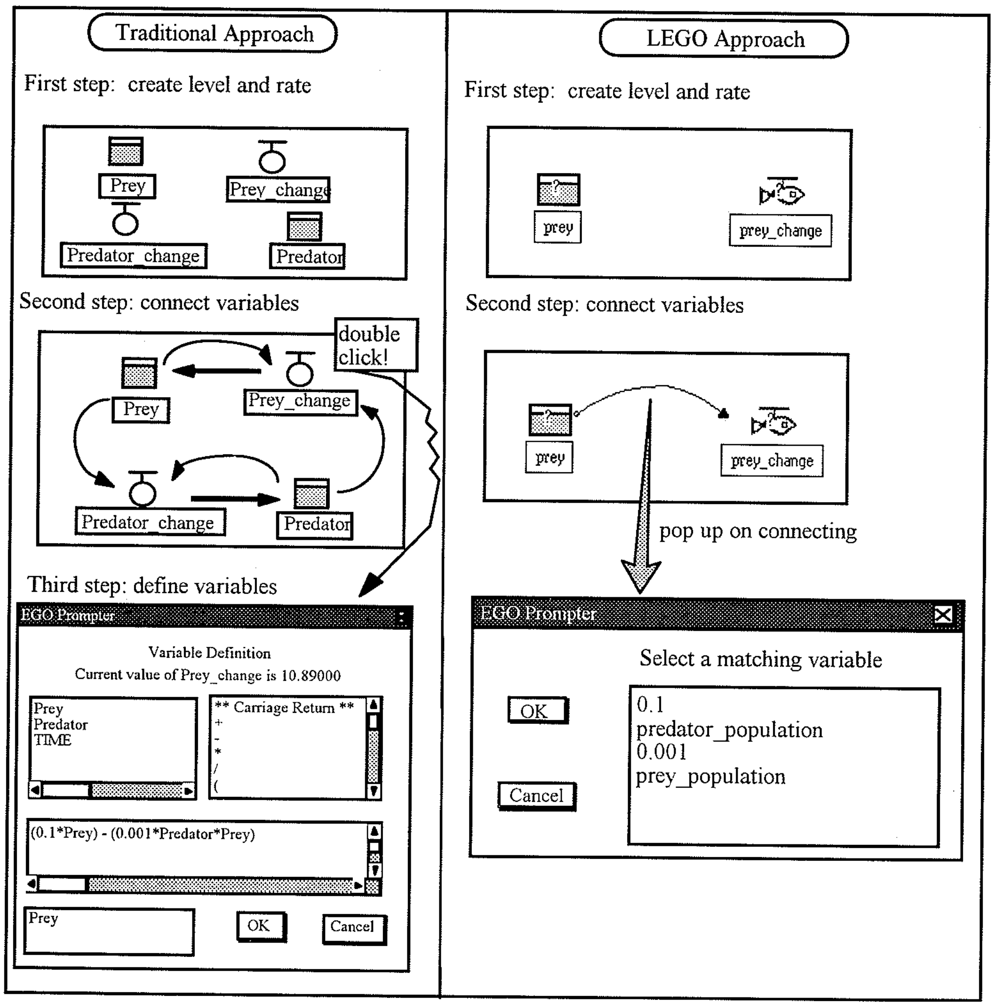


Figure 1. Reduced steps in LEGO modeling approach

LEGO in EGO

The LEGO approach is implemented into my system dynamics software called EGO (Equations as Graphic Object) that was made from object-oriented language (http://soback.kornet.nm.kr/~sddhkim). LEGO utility in EGO is most useful when a few formula repeat in many equations. A system composed of many agents with similar decision rules is also a promising candidate to model with LEGO in EGO (Resnick 1994).

Before playing LEGO in EGO, one must prepare user-defined icons. General guidelines for constructing a LEGO in EGO are as follows:

① Find basic building variables (repeating equations).

② Draw simple icons for depicting the variables.

③ Write prototype equations using explanatory names (For example, use "population of predator" rather than "predator")

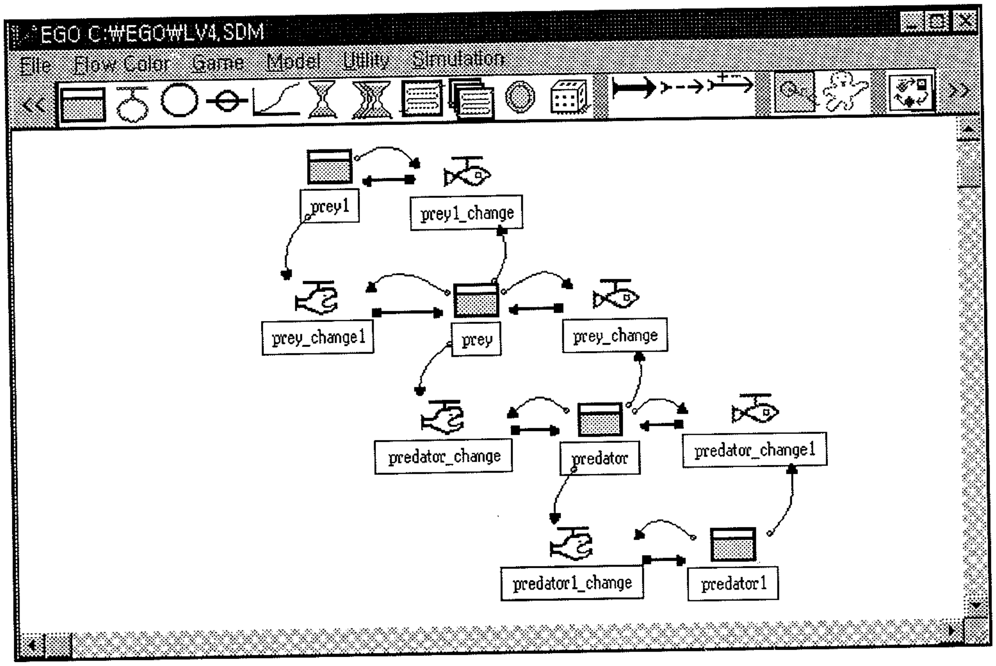
As an example of Lotka-Volterra equations, I defined two icons, one for the rate variable as a prey_change and one for a predator_change. Table 1 is a file for defining these user-defined icons. One can load this file by choosing Utility's submenu; 'New Icons'. In the file, one must record informations for new icons including its name, a file name containing its graphic form, a position in the icon menu, and its prototype equation.

Table 1. Contents of a file for user-defined icons

```
prey prey.dib 800 0 840 40
  (0.1*prey_population) - (0.001*prey_population*predator_population) !
predator predator.dib 850 0 890 40
  (0.001*prey_population*predator_population) - (0.1*predator_population) !
```

With user-defined icons in EGO, one can build a model within only two steps: create and connect. A user-defined icon knows its own prototype equation. All things students must do is to create and connect appropriate variables and to click one of the prototype variables that pops up on connecting variables.

With the LEGO utility in EGO, we can replicate a prey-predator system, and connect them into a whole system. Figure 2 is an example for using LEGO in EGO to build a double prey-predator model. One need not care about formulating equations at all. Just create, connect, and select.



Concluding Remarks

When we can construct a rich pool of user-defined icons, I believe that children can learn system dynamics modeling with fun. It seems that less than hundred user-defined icons will be sufficient for teaching system dynamics modeling at K-12 schools. And they can learn how to define variable equations after learning fundamental mechanism of system structures and having enough interest in the system dynamics modeling.

Reference

M. Resnick, 1994, *Turtles, Termites, and Traffic Jams*, The MIT Press.