*Annales Mathematicae et Informaticae*
*43 (2014) pp. 183–199*
http://ami.ektf.hu

# IT Competences: Modelling the Real World

**Péter Szlávi, László Zsakó**

Eötvös Lóránd University
szlavip@elte.hu
zsako@caesar.elte.hu

*Submitted January 23, 2014 — Accepted December 30, 2014*

**Abstract**

We often understand real-world phenomena through their models. To be able to do this, we need to be aware of the basics and the operation of modelling and the methods of its application. Modelling should be used not only for understanding but also for predicting real-world phenomena. The IT-specific peculiarity of modelling is that even the operation of models is a complex creative process.

*Keywords:* Model, Simulation, IT Competences, Elementary and Secondary School

*MSC:* 68U06

Having IT competences means that one is able to apply the basic IT tools and methods for getting information and solving problems in one's everyday life, both at home and at the workplace. People with such skills are able to use their knowledge practically, to learn and operate new technologies and methods, to solve problems, to reach individual and social goals, and to make informed decisions in our information society. [1, 2]

The important IT competences are the following[1]:

- Algorithmic thinking [3]
- Data modelling
- Modelling real-world phenomena
- Problem-solving

[1]The majority of these competences appear in other disciplines as well.

- Communication skills
- Application skills
- Teamwork, cooperation skills
- Creativity
- Information literacy
- Systemic thinking

Modelling is of course not an IT competence only. Models are used in several disciplines of science, such as mathematics (for phenomena like the point, which has no dimensions, the line, which has no thickness, or infinity), physics (for phenomena like friction or entities moving without air resistance).

Blum, referring to modelling as a mathematical competence, writes: *"Modelling competence includes the following: to structure, to mathematize, to interpret and to solve problems and it includes as well the ability to work with mathematical models: to validate the model, to analyze it critically and to assess the model and its results, to communicate the model and to observe and to control selfadjustingly the modelling process."* [4]

Nevertheless, three basic differences need to be pointed out that make IT special among the school subjects:

- In IT students can make models based on real-world systems (this is possible in mathematics, and only in very rare cases in physics, too [4, 5, 6, 7][2]);
- In IT models can even be made real with the computer (and often the realization is more important than the usage of the model);
- The very operation of IT models is a complex creative activity.

In other words, IT has a tool (the computer) for making models. As a consequence, the IT models are based on data (determined by the features of real-world objects), with which the computer algorithm performs calculations, and these results inform us about the specific real-world system. [8]

One of the most efficient tools of cognition is modelling. Those who are able to clearly describe a process they experience using abstract terms are on the right track to say: "We have understood the phenomenon." By operating the model, they can gather abstract experiences, which can serve as bases for real-world experiments. Those experiments then enable them to polish the model, thus, their knowledge.

Modelling is a "schematic process," which calls for the consideration of the following. First, we need to define the abstract objects of the model, which are the

[2]Gabriella Ambrus wrote the following about mathematical modelling: *"When solving a modelling task, the focus is put on the process and procedure(s) the student has to find and implement in order to create a relation between a non-mathematical problem and some mathematical content."*

"metaphors" of the objects (or classes of objects) in the real-world system. Later, we need to determine the set of states of the objects, (one of) which will characterize them during their operation. Finally, we set the rules of state change with the appropriate algorithms. [9, 10]

**Their levels:**

- Defining the models of reality
- Operating the models of reality
- Comprehending reality through the models
- Making the models of reality
- Learning model making
- Predicting real-world phenomena through the use of the models
- Analyzing the models

The competence levels of mathematical modelling is more thoroughly described by Herbert Henning and Mike Keune [11]:

*Level 1 – Recognize and understand modelling – is characterized by the ability:*
*to recognize and*
*to describe the modelling process,*
*to characterize, to distinguish and to localize phases of the modelling process.*

*Level 2 – Independent modelling – is characterized by the ability:*
*to analyze and to structure problems and to abstract quantities,*
*to adopt different perspectives,*
*to set up mathematical models,*
*to work on models,*
*to interpret results and statements of models,*
*to validate models and the whole process.*

*Level 3 – Meta-reflection on modelling – is characterized by the ability:*
*to critically analyze modelling,*
*to characterize the criteria of model evaluation,*
*to reflect on the cause of modelling,*
*to reflect on the application of mathematics.*

## 1. Where Do We Find IT Modelling in Public Education?

Modelling can occur in many forms.

## 1.1. Data Modelling

*"Data modelling is an abstraction process, in which the real-(micro)world facts and the data about the relations between these facts are collected and are converted in a format applicable for computer adaptation, that is, in so-called data models. Data modelling is concerned with the internal structure and relations of the data, but not with their specific values."* [12]

## 1.2. Problem-Solving: The Models of Problems

During the process of problem–solving, we need to define the model that describes the problem, as the initial step. [13] Let us demonstrate the series of steps of modelling through a task made for a qualifier for the student olympiad (in 2012).

*A farmer had three cans of milk of different capacity measures. When they are full, they have the capacity of A, B, and C liter of milk. The farmer knows that by pouring milk from one can to the other, it is possible to measure a specific quantity of milk. All the farmer needs to do is keep in mind how much milk is left in the one used for pouring and how much milk is filled in the one used to contain the pouring. In the beginning all the cans contain some milk, but the farmer wants to have all the milk for sale to be in can A. In order to minimize the time used, the farmer needs to know that the time spent on filling equals the amount of milk poured. Make a program that calculates the time and the amount of milk, the measurement of which takes the most time.* [14]

In the beginning the three cans contain $(a, b, c)$ liter of milk. We can only pour from a can that is not empty and pour into a can that is not full. After every pouring either the can, used for pouring, becomes empty, or the can, used for containing, becomes full. That is, a state $(a, b, c)$ can turn into a $(0, b + a, c)$ or $(a - (B - b), B, c)$ state, for example. The model of the problem, therefore, is a graph, whose points are the states describing the current states of the cans and the edges are the regular pourings.

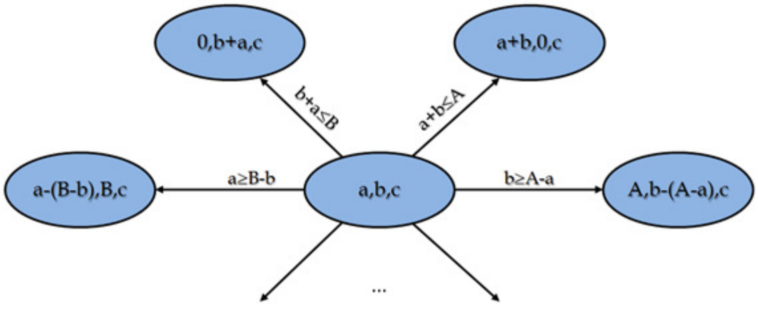

Figure 1: The graph model of the problem

The model of the problem is a data model. It is this onto which we can build a calculation model, that is, an algorithm, with whose implementation the problem can be solved in the case of any initial state.

## 1.3. Robots as the Models of Technical Systems

It can be a task even in grades 1 to 4 to move robots (cars, for example). These cars can resemble real cars, as they can move forward and backwards, or turn left and right. Nevertheless, they can be different as well: often their turning is not executed by manipulating a steering wheel but by switching the engine of the side wheels on and off.

In the case of managing toy trains and traffic lights, a very similar technical system model is in operation. As the main focus is on their operation, not on their structure, the robots can be models not only of technical but biological systems as well.

This issue is relevant in IT when we want to program robots. In such a case, a sensor of the robot converts the signs perceived from the environment into data, then our program makes a calculation, and the results will be used to control the robot.

## 1.4. Simulation Models

To understand the operation of real-world systems, we are often able to use simulation models. [9] Simulation models operate very similarly to real-world systems, but they function within a virtual world, not in reality, as opposed to the previous type. While technical models require us to build the model, simulation models require us to write a program.

What do we mean by computer simulation? Its essence, in short, is the following: it is the model of an examined universe (may that be biological, chemical, or relating to any other scientific discipline, including economic micro-worlds), which lies on the stochastic state changes of a discrete object and gets embodied in a program. In other words, the program is the tool which the users, that is, the experimenters hold in their hands to compare the ideas of the modeller with the facts of reality.

It is worth to note that simulation has a significant "advantage" compared to the regular, abstract tool of mathematical modelling. In the latter, a serious abstraction process is involved to obtain the mathematical variables, among which some kind of formal mathematical relation needs to be found, which is another abstraction process. (In addition, based on the number and relation of the different parameters and variables we need to recognize which specific mathematical subfield is to be utilized, starting from linear equations and equations systems, through the non-linear, to the ordinary and partial differential equation, or the tools of stochastic processes.) Contrary to this, simulation modelling "only" requires us to "copy" real-world relations. A further difference is that the model keeps the dynamics and naturalness of the real-world system; it is interaction that expands the possibilities of the model through the temporal changeability of the context.

## 1.5. Computer Model

A model that models the computer itself is useful because it can demonstrate and help understand its operation, of course if it adapts to the level of the audience. One such model is exemplified by [15]. Two "peculiarities" can be spotted in its adaptation:

1. The model in reality is a series of models, as it aims to demonstrate the real "evolution" of the computer, following the principle of historicity in the enumeration of the different computer models.

2. The models go beyond technical "novelties"; they touch upon several seemingly "secondary" features. They mention, for example, the programming process, the relevance and role of operation systems, and some of the problems of parallel calculation (in a level still comprehensible for kids).

## 1.6. Network Models

The physical manifestation of computer networks are partly covered by network topologies (ring, hub, etc.), which are basically graph models of the network structure. [16]

## 1.7. Communication Models

A classical network communication model is the ISO OSI (Open System Interconnection) model, which describes the relations of the specific machines within the computer systems on several levels. Even though today the protocols used are not directly connected to this model anymore, it is still very useful for understanding communication and reviewing tasks. [16]

## 1.8. Computer Games as Real-World Models

While in the case of simulation models, as analyzed in chapter 1.4, we can control the model from an omnipotent position, changing the parameters that characterize the modelled world, the genre of computer games grants us with a more or less cooperational position in shaping the modelled events. When talking about computer games, we mean both online, multi-peered applications and "regular" programs. Among the participants we can find real people, next to a compulsory artificial intelligence, who is either only a mediator of the "messages" of the human players or functions as a real playmate, fighting for the same goals.

These games are models of the real world, idealized for a certain goal. This goal can be to manage within a human community, but it can also be to survive within an abstract world organized around certain rules. To quote examples to the former type, we could list the myriads of online role-play games, for instance the well-known WoW (World of Warcraft). As the other extreme of computer games, we could mention programs simulating natural or social processes (SimEarth, SimCity, and Civilization, among others) or the computerized versions of "classical" board and card games.

Note that the above listed "social" games do resemble reality, but often times they lack the intention, characteristic of modelling, to mimic real conditions. In this sense, they are not models of reality. They are simulations, but the world we build for the simulation to take place is virtual (where events and objects are called the same as in the real world, and they might even look alike). Since the context is virtual, the results do not have to be related to real conditions. Despite this, the experiences are still useful.

Computer games are worth mentioning for two reasons. On the one hand, they are playful and as such highly motivating; thus, they can serve as specifically effective educational tools. On the other hand, they can be used as programming tasks. Adjusted to the programming competence of the students, we can set tasks related to the game, which students will be enthusiastic to solve given its playful nature.

## 2. The Methodology of Modelling

The **simulation models** and the **models of technical systems** (which also function to simulate operation) can be used within the subject of IT as the models of real-world processes. Therefore, we are going to deal with them when modelling the real world.

A **model** is a schematic notion made generally for understanding the operation of a complicated, not thoroughly known system, from which new correlations can be drawn, or which enables us to describe the phenomena of this system mathematically. The model usually reflects only the main features of the real-world system, in a simplified way. Which counts as the main features depends always on the purpose of the model. Therefore, by models we mean – mathematical – constructions that describe the observed phenomena. Such – mathematical – constructions are justified exclusively and exactly by their operation.

In the following we are going to use the notions of **modelling** and **simulation** always according to what aspects we want to refer to modelling from.

**Modelling**: the process of making a model. **Simulation**: the process of using a model.

The modern "model method" relates very closely to reality. The first usable and successful models were made in physics; such was the ideal gas, the perfect liquid, the point-like particle, the mathematical pendulum, or the atomic model. Physics was followed a lot later by chemistry, biology, and earth sciences.

The steps of modern model method [17]:

- *Gathering experiences through observation.*
- *Making a model to understand the experiences.*
- *Predicting the unknown phenomena with the help of the model.*

- *Checking the validity of the prediction with experiments and determining the validity limit of the model.*
- *Solving practical tasks with the help of the model within its validity limit.*
- *Developing, modifying, or replacing the model for understanding the phenomena beyond the validity limit.*

The model, naturally, needs to be monitored. What is even more important than this, however, is that it needs to be adjusted and changed if new aspects come up.

In the digital world, it is not surprising if we add the making of the computer model (that is, the program) to this and perform the prediction with the help of operating this computer model. Our activity is characterized by a process which consists of the following objects and operations [9]:

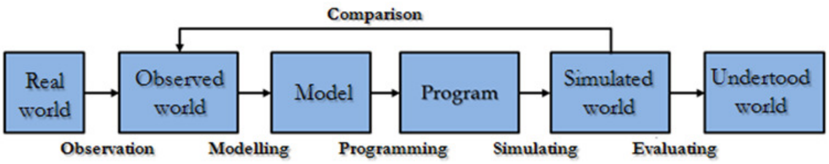

Figure 2: The process of understanding through simulation

Note: In a general sense the whole process can be called simulation. We can find a similar figure in the Werner Blum's (1996) article [18], which has been widely quoted ever since:

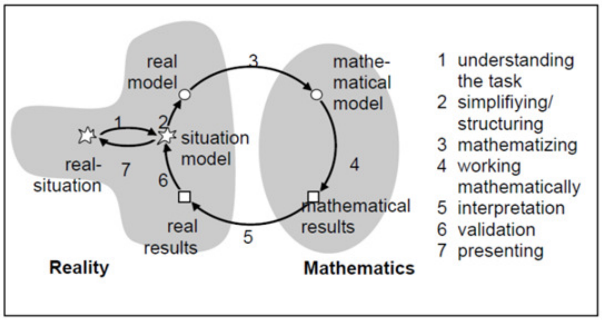

Figure 3: The process of mathematical modelling according to Blum

In every step of our activity we have to make sure that the features of the object created match those they refer to (for example, if the program has all the characteristics we expect from the model), and that the given objects themselves are valid. The final decision can be made after the comparison, which is when we can evaluate our entire activity. If the results of the simulation meet our expectations, we have reached our goal. If not, we have to check in which step we made a mistake, and correcting it, we go through the above described process again.

During modelling we use the observed data to describe the system with the help of – generally mathematical – methods, from which we can draw conclusions about the characteristics of the real-world system; we can even predict its behavior. There exist numerous examples to such formalisms (like ordinary or partial differential equations, difference equations, finite state machines, or Markov chains), but as they involve complicated mathematical apparatuses, we avoid using them in public education.

**The model itself can never be the goal of modelling.** The model needs to be suitable for the analysis of the modelled; thus, it must be functional and operational. Before moving on to the next activity, we always need to make sure the model fits our goal. If we apply some mathematical method, we need to check, by solving the mathematical task or providing a partial solution, if we get the expected results (which is why it was important to set the expected results in advance).

If the goal of our model is **demonstration**, for the sake of the result it is acceptable for the internal structure of the model to have some relations that are certainly different from the real phenomenon, because in this case only the result matters. (Naturally, the relations must not concern the essential mechanism of the model's operation; that is, we cannot suggest false ideas.) If, however, the goal is some kind of **analysis**, the model must follow our conceptions of the real-world system, avoiding "false analogies" (which is how the term of *horror vacui*, suggesting that bodies fall downwards due to a fear of the empty, was born).

Before making the model, we observe and gather information, formulating hypotheses about the modelled system, regarding its objects, their relations, their states, their changes, the external forces, and the overall state of the modelled system.

In the case of regular mathematical models, creating the model means defining the mathematical correlations between the parameters and state changes of the system, while for constructive methods there are other – easier – ways.

As the first step of model making, we need to define the abstract **objects** of the model, which correspond to the objects (or classes of objects) in the real-world system. This correspondence usually involves the correspondence of their states as well. To be able to speak about them on their own (like in the real-world system), they require individual existence, which is replaced by setting their state. As the next step, we need to make the algorithm describing the **state changes** (change in number, change in state) of the objects.

The significant difference from mathematical models lies in the circumstance that it is not the mathematical variables, defined as a result of serious abstraction efforts, that we need to find a mathematical relation for (which is another abstraction process). Our task is to "copy" the relations of the much more easily understandable real-world system. A further difference is that the model keeps the dynamic structure, results, and naturalness of the real world.

It is a very frequent mistake in model making that our observations are inadequate, inaccurate, and not goal-oriented enough. This actually is an inevitable obstacle in every case; therefore, we need to address it. If we refuse to acknowledge these inadequacies and inaccuracies, we can cause great damage to ourselves and – if the goal of modelling is demonstration – to others too.

The algorithm correctly describes the operation of the real-world system if:

- we take a random state of the system,
- we do the correspondences in the model,
- we create a future state of the model with the help of the algorithm,
- we find the real-world correspondent of the model state (result),
- and the result "matches" the real-world state (in deterministic cases it means equation, while in stochastic cases it is equal distribution).

The model–modelled relation is similarity: the model is similar to the modelled only from a certain aspect. The existence of such a similarity is crucial, since this guarantees that by knowing the state of the model we can define the state of the modelled too (certainly or with great probability – statistic models).

Of "whole models," resembling the modelled completely, there exists only one; the modelled itself. As a consequence, we always need to determine from which aspect the model needs to resemble the modelled. It can also occur that modelling is possible only through a very rough approximation, but this, as Hans-Wolfgang Henn points out, is not a problem. "*Models for a real problem can be more or less suitable. One should never talk about 'right' or 'wrong'. For example, it does not make sense to call Newton's model of physics 'incorrect' and Einstein's model 'correct'.*" [19]

It is an important quality that the above similarity is equivalence relation, mathematically speaking. As a consequence, the model of the model is the model of the modelled too, provided that we followed the same principles when making it. Like this, we can guarantee, even in the case of a long abstraction process, that we are still talking about reality. Another expected quality is that the modelled is the model of its own model, which means that the relation is valid only if the events of the model can take place in the real-world system as well.

## 3. Modelling for Different Age Groups

Development of the models: the older the students are, the more realistic, the more profound, and the more structured the models in use will be.

## 3.1. Tasks for Grades 1 to 4

The models of the first four grades come from the world of games: Lego, toy cars, and model railways are natural models for children.

In other words, models of technical systems (like Lego robots) can be used for this age group. These models can be grouped into two major types: moving robots or robots parking objects.

Both types can be controlled, first without any perception of the external world. As a second step, we can equip them with sensors, so we make their operation dependent of some external condition (for example, the car should not bump into the wall; the car should follow a path on the floor; the robot that parks cars should load objects of certain height on top of each other).

The shared characteristic of these models is that they are all programmable. Programmability, however, can mean a simple puzzle game, in which we graphically edit the program units together in order to form the appropriate program. [20]

## 3.2. Tasks for Grades 5 to 6

The modelling of reality, suitable for this age group, can be of two kinds. On the one hand, we can continue the application of robots as the models of real-world systems, within more complex tasks, possibly involving programming solutions as well. Our model (in this case, the robot) operates within a real-world environment, reacting to all changes and events of it. Nevertheless, the robot can be perceived like a limited automaton: it is exposed to the effects of the environment, to which it reacts with the change in its state components; this, in turn, affects its environment.

On the other hand, computer simulation models, more distant from reality, can also appear; their function is to simulate some kind of phenomenon with the help of the computer. To be able to do this, the application of random number generators is indispensable. The computer modelling of simple random phenomena (like dice, screen-walk, and random figures) becomes possible.

Note that we consider it important that the simulation of random phenomena, or even more random games, precedes the simulation of non-random events.

Here the simulation model does not relate to a real-world system, at least not directly.

## 3.3. Tasks for Grades 7 to 8

The models, used for understanding the operation of real-world systems, not for the joy of playing, can come to the foreground with this age group. We can even begin to introduce the computer modelling of simple random – and non-random – natural phenomena.

For this age group, the model becomes the simplified copy of reality (as we try to mimic reality as closely as possible); we deal primarily with defining the model and realizing it with the help of the computer.

Extending the tasks with robots requires us to introduce o new terminology, namely, the "model of the model." It is difficult – not only time-consuming but occasionally even dangerous – to test complicated programs with robots. This is when it is useful to simulate robots with the help of the computer. Only when we tested several types of robot control and eliminated the occurring mistakes can we begin to test the finalized program with the robot. In other words, the robot is a simplified model of a real-world object, while the robot simulation program is the model of the model.

Side-note: In the case of both controlling robots and simulating natural phenomena, the structure of the algorithm (program) varies significantly from what we are first introduced to in the classical programming education. Traditionally, we assume a deterministic, sequential implementation, whereas robots and interactive simulations react to the effects coming from the environment. According to the classical program structure, the program is composed of three elements: scanning data, calculating results, and showing results. Both robot control and simulation are based on online algorithms, which means that we obtain and communicate new data during the calculation.

According to Eigen and Winkler's book, we can interpret the computer simulation of natural phenomena like a board game: *"Play is a natural phenomenon that has guided the course of the world from its beginnings. It is evident in the shaping of matter, in the organisation of matter into living structures, and in the social behaviour of human beings.... Every game has its rules that set it apart from the surrounding world of reality and establish its own standards of value."* [21] The elements of the real-world systems we want to model can be grouped into classes, and the number of these elements (and sometimes the spatial distribution and pattern of the classes) determines the state of the system.

The specific units of the system can be in a finite state (discrete state model), and the number of units can also be only finite. The future (next moment) state of each unit is determined by the current state of the elements and the parameters of the system. We need to store information about the specific units, for which tables would be appropriate. If the spatial location of the units is irrelevant, we can use vectors; if the physical proximity of the units is relevant, however, two-dimensional tables (matrices) would be the better choice. We need to fill out the table based on some initial distribution, generally randomly. After this the changes, that is, the events can start.

We can define the future state deterministically, provided that we have a direct rule for calculation, or randomly. In the latter case, several calculation rules can be assigned to the specific units, coupled with a calculation rule about how likely its application should be.

The challenge of translating the parallelism of real-world systems into the sequential von Neumann computer can be overcome in two ways:

- Event stepping: we constantly monitor the on-going events. In a given moment one (or more) event(s) can occur. The question is how we choose that one event.

- Time stepping: we monitor the events with discrete intervals. In a given moment every unit of the system is "in an action." The challenge is to set an adequate order to the events.

It could be interesting for students of the Logo programming languages to get acquainted with the simulation environment of NetLogo. [21]

## 3.4. Tasks for Grades 9 to 10

For this age group, the scientific simulation can extend and cover all subjects of science, including biology, chemistry, physics, and geography. The obvious question arises: Which subject should deal with such simulations? Our answer is that both IT and the specific science subject. To explain IT's part we should note that simulation applies general patterns which are independent from the specific discipline, such is Eigen and Winkler's classical book [22]. In essence, its realization is primarily a programming task.

In addition to the simulation of random phenomena, we can also introduce the simulation of deterministic events too. As a start, it is wise to choose phenomena whose temporal display is spectacular. A possible example for a very simple task is to move a point on the screen with a given speed in a given direction, make the point bounce off at the edges of the screen, due to friction make it get slower and slower, apply the downward force of gravitation... and the instructions can go on.

The idea behind such simulations is that we have a given number of elements, with a given set of characteristics about their motion. We break up the time into smaller units and we calculate each element's state in the next time unit. We delete the previous position of the elements from the screen and draw them in their new one. And then comes the next time unit. (This approach can lead us also to simulation games, where, for example, a player has to catch the point, described in the previous task, using some kind of tool.)

The operation of models can serve the goal not only of understanding but also of experimenting ("what if"). [23] This type of simulation, however, belongs more to the specific discipline, not IT.

The realization of the model, with the help of the computer, is perhaps more important for this age group than experimenting with the help of the model. Experimentation becomes much more enjoyable with simulation games.

## 3.5. Tasks for Grades 11 to 12

It is worth continuing with subject-specific simulations but primarily in classes where students specialize in the given subject, studying it in higher weekly hours.

For this age group, two subjects, linked much more closely to real-world applications, can appear in the curriculum: Simulation of transportation systems; EconoNumbermic simulation.

Interestingly enough, it is very often the case that these grow out of the world of computer games (skill-based games, car race, strategic games).

Furthermore, prediction can be introduced as another goal of simulation for 11th and 12nd graders. An example for this is the demographic simulation based on the Leslie matrix we can see below.

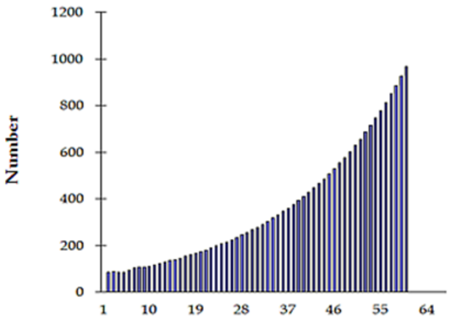

Figure 4: Demographic prediction with the help of the Leslie matrix

The above figure, just like the one below, shows that we can skip illustrating the "participants" of the simulation.

A very interesting experiment can be shown to this age group: the use of special software systems, not necessarily developed but applicable for simulation (take GeoGebra [25, 26] and spreadsheet [27] for example).

The following task, the simulation of the vibration of an object with M weight, on an "ideal" spring of 0 weight and D elastic modulus, can be easily solved with spreadsheet. We know the initial deflection ($s_0$), the acceleration of gravity, and the drag coefficient (d). The simulation's principle is that if we choose short dT intervals (assuming that the changes of the parameters are negligible during these dT intervals), the important state variables can be easily calculated and (for example) a state diagram can be drawn:

$F$ **force**: $F = F_s + F_d + G$, $F_s = D \cdot s$ (*spring*), $F_d = k \cdot v$ (*drag coefficient*), $G = M \cdot g$

$a$ **acceleration**: $a = \dfrac{F}{M}$

$v$ **speed**: $v = a \cdot dT$

$s$ **deflection**: $s = v \cdot dT$

$s_0$ **initial deflection**: $s_0 = \dfrac{M \cdot g}{D} \Leftarrow M \cdot g = s_0 \cdot D$

The relevance of modelling is well demonstrated by the fact that there have been robot programming competitions, organized for high school students, for years now,

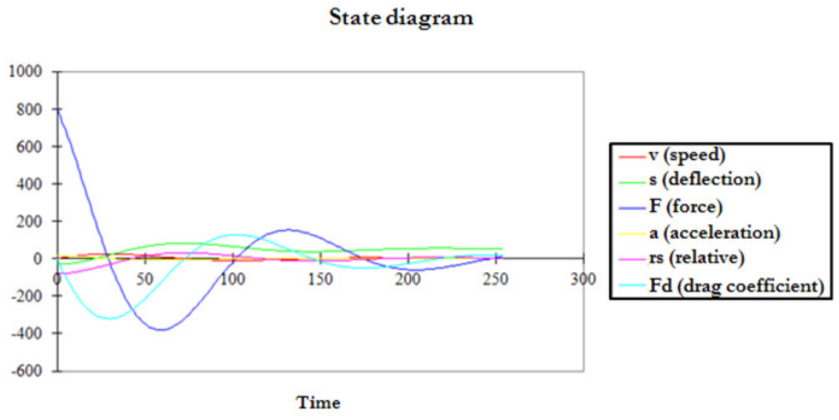

Figure 5: State diagram about spring motion

and even traditional programming competitions embrace more and more simulation tasks.

At the Imre Gyula Izsák Mathematics–Physics–Informatics Competition, initiated in 1992, it has been a practice to include, as part of the IT assignment, a simulation task connected to physics (for example, motion in the gravitational field – 1995, refraction and reflection – 1996, and so on). [28]

In 2012 even the two major national competitions in IT (Nemes Tihamér OITV, OKTV) introduced the first simulation tasks (transportation simulations - intersection and pedestrian crossing), which, despite their novelty, not only became the favorites of the competitors, but they were also solved successfully by many). [29]

## 4. Conclusions

The question might arise why it is the IT classroom that has to make room for all these, and why it is IT teachers, not physics, biology, literature, etc. teachers, that have to teach the above described skills. We may answer this question, partially, with a cultural historical analogy, which we attribute to Győző Kovács:

*The Christian religion spread not all by itself, and it is not the Roman Pope or the 20–30 bishops whose role was the most significant in its dissemination. Instead, it was thanks to the small chapels and missionaries that the religion, with the related technological and cultural knowledge, reached every village. The "missionaries" of the IT applications are the IT teachers; only they can be able to convince the other 95% of the teachers about the possibility and the need to incorporate IT in the wide range of school subjects.*

On the other hand, the majority of the models can be schematized. It means that we can make model frames (in a more trendy word, templates) that help shorten and simplify the modelling process. In addition to this, templates can enable us to categorize models according to their qualitative behavior; we can distinguish, for example, specific basic models and basic growth models. [9, 21] We believe model making, especially computer-based model making, is an important and clearly IT field; consequently, it belongs in IT education.

Computer *simulation* can lead to monumental tasks, which often require serious discipline-related knowledge as well. As a consequence, it facilitates project work in larger groups, where both IT competence and discipline-specific knowledge are needed.

## References

[1] Vass, V. (ed.); Kompetenciaháló (Competence Web), *Nemzeti Tankönyvkiadó*, (2009).

[2] Szlávi, P.; Zsakó, L.; ICT competences – Algorithmic thinking, *Acta Didactica Napocensia* Vol. 5. No. 2. (2012), pp. 49–58.

[3] Horváth, Gy.; Szlávi, P. and Zsakó, L.; Informatics (ICT) competencies, *ICAI 2010 – 8$^{th}$ International Conference on Applied Informatics Eger, Hungary, January 27–30*, (2010).

[4] Blum, W.; ICMI study 14: Application and Modelling in Mathematics Education–Discussion Document, *Educational Studies in Mathematics* 51, (2002), pp. 149–171.

[5] Ambrus, G.; Modellezési feladatok a matematikaórán (Modelling Tasks for Maths Classes), *Matematika–Tanári Kincsestár, B 1.2, RAABE Tanácsadó és Kiadó Kft., Budapest*, (2007).

[6] Tóth,B.; Modellezési feladatok a matematikában (Modelling Tasks in Maths), *ELTE TTK*, (2010).

[7] Vancsó, Ö.; A matematikai modellezés nehézségei egy 2009-es OKTV feladat kapcsán (The Difficulties of Mathematical Modelling Based on a Task in the National IT Competition), *A Matematika Tanítása*, 2009/9, *Mozaik Oktatási Stúdió, Szeged*, (2009), pp. 30–34.

[8] Abramovich, S.; Spreadsheet-Enhanced Problem Solving in Context as Modeling, *Spreadsheets in Education (eJSiE)*, Vol. 1 Iss. 1 Article 1, (2003).

[9] Horváth, L.; Szlávi, P. and Zsakó, L.; Modellezés és szimuláció. (Modelling and Simulation), *Mikrológia 1. ELTE IK*, (2004).

[10] Szlávi, P.; Zsakó, L.; Az informatika alkalmazási típusai a közoktatásban (The Application Types of IT in Public Education), *Informatika a Felsőoktatásban'96 – Networkshop'96, 1996 August 27–30*, (1996), pp. 534–543.

[11] Henning, H.; Keune, M.; Levels of modelling competence – Modelling and Applications in Mathematics Education, *New ICMI Study Series*, Vol. 10, (2007), pp. 225–232.

[12] Bércesné Novák, Á.; Az adatmodellezés szintjei (The Levels of Data Modelling), (2013)
Available at `http://www.inf.u-szeged.hu/oktatas/Tempus/abkr2.doc` (accessed 31 December 2013).

[13] Geda, G.; Hernyák, Z.; Algoritmizálás és adatmodellek (Algorithms and Data Models), *Kempelen Farkas Hallgatói Információs Központ*, (2013)
Available at `http://www.tankonyvtar.hu/hu/tartalom/tamop425/0038_informatik a_Geda_Gabor_Hernyak_Zoltan-Algoritmizalas_es_adatmodellek/ch03.html` (accessed 31 December 2013).

[14] Informatikai Diákolimpiák válogató versenye. (Qualifiers for the Olymiads in Informatics), (2013)
Available at http://tehetseg.inf.elte.hu/valogatok/valogatok_main.html (accessed 31 December 2013).

[15] Szlávi, P.; A számítógépről népszerűsítő stílusban (About the Computer in Promotional Style), *Mikrológia 5. ELTE TTK Informatikai Tanszékcsoport*, (1988).

[16] Tanenbaum, A. S.; Computer Networks, *Pearson Education, Inc. Publishing as Prentice Hall PTR Upper Saddle River, New Jersey*, (2003).

[17] Dr. Huzsvai, L.; Kutatói pályára felkészítő akadémiai ismeretkörön alapuló tananyagfejlesztés – Környezet- és természetvédelem ismeretkörben. (Curriculum Development for Researchers – Nature and Environment Protection), (2013)
Available at http://www.tankonyvtar.hu/hu/tartalom/tamop425/0032_kornyezetved_termved_kutatoi/ch02.html (accessed 31 December 2013).

[18] Blum, W.; Anwendungsbezüge im Mathematikunterricht – Trends und Perspektiven, *Schriftenreihe Didaktik der Mathematik*, Vol. 23, (1996), pp. 15–38.

[19] Hans-Wolfgang Henn; Modelling in school – chances and obstacles, *The Montana Mathematics Enthusiast*, ISSN 1551-3440, Monograph 3, (2007), pp. 125–138.

[20] van Lith, P.; Teaching Robotics in Primary and Secondary schools, *ComLab Conference 2007, November 30 – December 1, Radovljica, Slovenia*, (2007)

[21] Wilensky,U.; NetLogo, (1999)
Available at http://ccl.northwestern.edu/netlogo (accessed 31 December 2013).

[22] Eigen, M.; Winkler, R.; A játék (The Game), *Gondolat*, (1981).

[23] Interactive Science Simulations, *University of Colorado at Boulder, PhET project*, (2013)
Available at http://phet.colorado.edu/ (accessed 31 December 2013).

[24] Age Structured Leslie Matrix, *Virtual Amrita Laboratories Universalizing Education. Amrita Vishwa Vidyapeetham University*, (2013)
Available at http://amrita.vlab.co.in/?sub=3\&brch=65\&sim=183\&cnt=1 (accessed 31 December 2013).

[25] Geda, G.; Bíró, Cs. and Kovács, E.; Számítógépes szimuláció GeoGebrával (Computer Simulation with GeoGebra), *INFODIDACT 2011, Szombathely, 2011. 03.31–04.01.*, (2011).

[26] Geda, G.; Bíró, Cs.; Tánczos, T.; Számítógépes szimuláció lehetőségei (The Possibilities of Computer Simulation), *Agria Media 2011, Eger, 2011.10–12.*, (2011), pp. 426–431.

[27] Szlávi, P.; Zsakó, L.; Szimulációs modellek táblázatkezelővel (Simulation Models with Spreadsheets), *INF.O.'97 Informatika és számítástechnika tanárok konferenciája, Békéscsaba, 1997. November 20–22.*, (1997).

[28] Izsák Imre Gyula komplex természettudományi verseny (Imre Gyula Izsák Complex Science Competition), (2013)
Available at http://www.zmgzeg.sulinet.hu/izsak/ (accessed 31 December 2013).

[29] Nemes Tihamér Országos Informatikai Tanulmányi Verseny – Programozás kategória. (Tihamér Nemes National Competition in Informatics – Category of Programming), (2013)
Available at http://tehetseg.inf.elte.hu/nemes/index.html (accessed 31 December 2013).