Title: The impact of the Computational Inquiry Based Experiment on Metacognitive Experiences, Modelling Indicators and Learning Performance

Article Type: Research Paper

Keywords: Metacognition, Modelling, Computational Experiment, Inquiry Based Science Education

Abstract: Computational experiment approach considers modelling as the essential feature of Inquiry Based Science Education (IBSE), where the model and the computer take the place of the "classical" experimental set-up and simulation replaces the experiment (Landau et al., 2008). Modelling, as a pedagogical tool, involves the model construction, the exploration of model characteristics and the model application to a specific problem, resembling authentic activities of scientists and mathematicians (Herbert, 2003).

Recent developments in strategy instruction research suggest that learning in a particular discipline is enhanced by guiding students through the development of content-relevant metacognitive strategies (Wosnitza & Volet, 2009).

Problem-solving is a complex process, which involves several cognitive operations such as collecting and selecting information, heuristic strategy and metacognition (Garofalo & Lester, 1985; Schoenfeld, 1994; De Corte, 2003).

The purpose of this study was to explore the impact of the Computational Experiment Methodology on learners' cognitive performance, use of modelling indicators and shift of the metacognitive experiences during problem solving using computational models.

Sixty prospective primary school teachers volunteered to participate in the study.

Students were exposed by the Instructor to a number of computational experiments, while during the course they developed their own models of simulation.
The results of the experiment show that the use of the computational experiment approach has a substantial effect on the metacognitive experiences and the use of modelling indicators.
I followed the comments

Sample is N=60, they participated voluntarily and the total number of students was N=79
The impact of the Computational Inquiry Based Experiment on Metacognitive Experiences, Modelling Indicators and Learning Performance

Abstract

Computational experiment approach considers modelling as the essential feature of Inquiry Based Science Education (IBSE), where the model and the computer take the place of the "classical" experimental set-up and simulation replaces the experiment (Landau et al., 2008). Modelling, as a pedagogical tool, involves the model construction, the exploration of model characteristics and the model application to a specific problem, resembling authentic activities of scientists and mathematicians (Herbert, 2003).

Recent developments in strategy instruction research suggest that learning in a particular discipline is enhanced by guiding students through the development of content-relevant metacognitive strategies (Wosnitza & Volet, 2009).

Problem-solving is a complex process, which involves several cognitive operations such as collecting and selecting information, heuristic strategy and metacognition (Garofalo & Lester, 1985; Schoenfeld, 1994; De Corte, 2003).

The purpose of this study was to explore the impact of the Computational Experiment Methodology on learners’ cognitive performance, use of modelling indicators and shift of the metacognitive experiences during problem solving using computational models.

Sixty prospective primary school teachers volunteered to participate in the study. Students were exposed by the Instructor to a number of computational experiments, while during the course they developed their own models of simulation.

The results of the experiment show that the use of the computational experiment approach has a substantial effect on the metacognitive experiences and the use of modelling indicators.

Keywords: Metacognition, Modelling, Computational Experiment, Inquiry Based Science Education

1. Introduction

The Computational Experiment

Over the past decade, increasing importance and attention has been attached to the potential of new technologies of information and communication (ICT) to improve teaching and learning in schools (Barton & Haydn, 2006). Focus on modelling is largely due to the following reasons. One is the recent constructivist attention to conceptions that students bring to the classroom. A second one is the present emphasis on the role of philosophy in science education, which has resulted in stressing the importance of attention for the nature of scientific models. Directly connected to that approach, is that proposed by Sloot (1994) as Computational Physics (CP). One of the crucial components of CP is the abstraction of a physical phenomenon to a conceptual model and its translation into a computational model that can be validated. This leads us to the notion of a computational experiment where the model and the computer take the place of the 'classical' experimental set-up, and simulation replaces the experiment as such. Sloot (1994) identifies three major phases in the process of the development of a computer experiment:

a. The modelling phase: the first step to simulation is the development of an abstract model of the physical system under study, b. The simulation phase: this refers to methods that make
the underlying physical models discrete in time or stochastic using methods from numerical analysis and c. The computational phase: in this phase we concentrate on the mapping of the simulation techniques to source code including algorithms and the implementation using software or a programming language.

Landau et al. (2008) suggests an approach similar to Sloot’s approach, which takes the form:

a. Problem (from science), b. Modelling (mathematical relations between selected entities and variables), c. Simulation Method (time driven, event driven, stochastic), d. Development of algorithm, e. Implementation of the algorithm (using Java, Mathematica, Fortran etc) and f. Assessment and Visualization exploration of the results and comparison with real data

According to Landau et al. (2008), Computational Physics (CP) and Computational Experiment (CE) provide a broader, more balanced, and more flexible education than a traditional physics major. Moreover, presenting physics within a scientific problem solving paradigm is a more effective and efficient way to teach physics than the traditional approach. In this framework, being able to transform a theory into an algorithm requires significant theoretical insight, detailed physical and mathematical understanding and a mastery of the art of programming and the actual debugging and testing. The organization of scientific programs is analogous to experimentation, with the numerical simulations of nature being essentially virtual experiments. The scientific paradigm should include modelling and simulation as an additional dimension in order to create computational experiments.

Tobochnik and Gould (2008) argue that CP should be incorporated into the curriculum because it can elucidate the physics. Computation is both a language and a tool and, in analogy to models expressed in mathematical statements, CP models are expressed as algorithms which in many cases are explicit implementations of mathematics. An advantage of the computational approach is that it is necessary to be explicit about which symbols represent variables and which represent initial conditions and parameters, boundary conditions, restrictions of the solutions etc.

According to Hestenes (1999), traditional physics courses lay heavy emphasis on problem solving and these results on the undesirable consequence of directing student attention to problems and their solutions as units of scientific knowledge. Modelling theory indicates us that these are the wrong units; the correct units are the models. Problem solving is important, but it should be subservient to modelling. Hestenes (1999) states that most physics and generally science/engineering problems are solved by constructing or selecting a model, from which the answer to the problem is extracted by model-based inference. In a profound sense the model provides the solution to the problem. Thus, an emphasis on models and modelling simplifies the problem and organizes a physics course into understandable units.

According to Hestenes (1999), model specification is composed by a model which describes (or specifies) four types of structure, each with internal and external components, considered as modelling indicators:

1. systemic structure which specifies composition (internal parts of the system), environment (external agents linked to the system), connections (external and internal causal links)
2. geometric structure which specifies position with respect to a reference frame (external geometry), configuration (geometric relations among the parts)
3. temporal structure which specifies change in state variables (system properties), descriptive models represent change by explicit functions of time, causal models specify change by differential equations with interaction laws
4. interaction structure which specifies interaction laws expressing interactions among causal links, usually as function of state variables

Shunn and Klahr (1995) and (Klahr & Dunbar, 1998) in order to describe discovery learning as a search process, introduced spaces in scientific discovery learning which include
the hypothesis space and the experimental space. In their model the hypothesis space contains all rules and variables describing the specific domain, while the experiment space consists of all experiments that can be implemented within this domain. Psycharis (2011) extended these spaces in order to include the computational experiment approach and suggested three spaces for the computational experiment, namely:

1. The hypotheses space, where the students in cooperation with the teacher, decide, clarify and state the hypotheses of the problem to be studied, as well as the variables and concepts to be used and the relations between these.

2. The experimental space, where the computational experiment actually takes place and includes modelling-simulation-based on discovery/inquiry learning activities for the problems under study, replaces the classical experimental and learners are engaged in the scientific method.

3. The prediction space, where the results, conclusions or solutions formulated in the experiments space, are checked with the analytical (mathematical) solution as well as with data from the real world and help learners towards predictions for other phenomena.

**The Computational Experiment (CE) and Inquiry Based Science Education (IBSE)**

The publication of the "Science Education Now: A renewed Pedagogy for the Future of Europe" report (Rocard, 2007), once again brought science as inquiry to the top of educational goals. The field of science education research is concerned –among other issues– with the development of high-level skills, like concept formation, modelling, problem solving, meta-cognitive skills and scientific procedures.

Inquiry based learning has been officially promoted as a pedagogy for improving science learning in many countries (Bybee et al., 2008; National Research Council, 2000). Inquiry can be defined as "the intentional process of diagnosing problems, critiquing experiments, distinguishing alternatives, planning investigations, researching conjectures, searching for information, constructing models, debating with peers, and forming coherent arguments" (Bell, Hoadley & Linn, 2004) and is often considered as a way to implement in schools the scientific method (Levy et al., 2010; Levy & Petrilis, 2011). Inquiry is referred to the science education literature to designate at least three distinct but interlinked categories of activity: (a) what scientists do when they use scientific methods, (b) how students learn (by pursuing scientific questions and engaging in scientific experiments by emulating the practices and processes used by scientists); and (c) a pedagogy, or teaching strategy, adopted by science teachers when they design learning activities, which allow students to observe, experiment and review what is known in light of evidence (Minner, Levy, & Century, 2010).

Developing inquiry-based learning environments with the integration of ICT seems to be an essential research issue in science education (e.g., Van Joolingen et al. 2005). Bell et al. (2010), identified nine main science inquiry processes supported by different computer environments that could be used in inquiry-based science education (IBSE), namely: orienting and asking questions; generating hypotheses; planning; investigating; analyzing and interpreting; exploring and creating models; evaluating and concluding; communicating; predicting. The nine inquiry tools of (Bell et al, 2010) are closely related to the essential features of Inquiry (Assay & Orgill, 2010), namely Question (Learner engages in scientifically oriented questions), Evidence (Learner gives priority to evidence), Analysis (Learner analyses evidence), Explain (Learner formulates explanations from evidence), Connect (Learner connects explanations to scientific knowledge), Communicate (Learner communicates and justifies explanations) and Reflection(Learner reflects on the inquiry process, respond to his/her work).
There is a close relation between the essential features and the nine tools of inquiry (Bell et al., 2010). For example, The “Question” as inquiry feature is related to the inquiry tools orienting and asking questions and generating hypotheses. “Evidence”, “Analyze” and “Explain” features are related to Planning-Investigating, Analysis and Modelling inquiry tools respectively and the features “Connect” and “Communicate” are related to the inquiry tools Conclusion – Evaluation -and Prediction. We observe that the role of modelling is essential both as an inquiry tool and as a feature of inquiry and since the Computational experiment has as fundamental component the development of models, its role seems pervasive in all inquiry processes. Our proposal integrates IBSE and CE through the interconnection of the CE spaces, namely the hypotheses space, the experimental space and the prediction space with the essential features of inquiry and the inquiry tools. In Table 1 below, we present the correspondence between the spaces of the CE and the features and tools of inquiry learning approach.

Table 1. The correlation of the CE spaces, the inquiry features and the inquiry tools

In Figure 1 we present the process of implementing CE in alignment with the previous discussion. The process consists of the cycle A-B-C where the different inquiry features correspond to the spaces of the computational experiment (CE).

*Figure 1. The process of the Computational Experiment combined with the Inquiry Based Teaching and Learning Approach*

*Metacognition in Science Education and Computational Science*

Although the concept of metacognition has been defined in numerous ways, Sperling et al. (2004) suggest a focus on its component parts, which are self-appraisal, knowledge about cognition and regulation of cognition. Knowledge about cognition refers to the level of the learner’s understanding of his/her own memories, cognitive system, and the way he/she learns; regulation of cognition refers to how well the learner can regulate his/her own learning system, i.e., goal setting, choosing and applying strategies, and monitoring his/her actions. Cognitive self-regulation refers to students being actively and purposefully engaged in their own learning. This includes analysing the demands of a learning task, planning and allocating resources to meet the task demands, and monitoring one’s progress towards completion of the task (Pintrich, 1999; Fernandez – Duque et al., 2000).

Literature has established the importance of metacognition in the acquisition and application of learning skills (e.g. Flavell, 1976; Panaoura, et al., 2009; Alexander et al., 2003). Metacognition is associated with planning, monitoring, evaluating and repairing performance, while metacognitive strategies guide students to think before, during, and after a problem solution. It begins by guiding students to plan for selecting the appropriate strategy to accomplish the task, and then continues as they select the most effective strategy and finally students evaluate their learning process and outcomes.

Research over the past three decades indicates that metacognition has been recognized as one of the most relevant predictors of accomplishing complex learning tasks in problem solving (Van der Stel & Veenman 2010; Dignath & Buttner 2008). Research on the reasons that students have difficulty in science, particularly physics, reveals that is not that they are too
young or lack intelligence, but rather that they simply do not know how to construct conceptual models of scientific phenomena and how to monitor and reflect on their progress (e.g. White & Gunstone, 1989).

Involvement of students in modelling is considered as a condition for their engagement in inquiry learning (Bell et al., 2010). Involvement in modelling can be implemented through programming, but claims that programming alone might improve problem-solving skills or other general metacognitive skills have gone mostly unsupported (Palumbo, 1990). Education researchers are more interested in programming as a medium, as a way of thinking about and exploring disciplines other than computer science (diSessa, 1991; Guzdial, 1991; Soloway, 1993). According to (Harel & Papert, 1990) it is useful to engage students learn about programming, because programming is an important skill and a medium of communication, but more important is to engage students in learning through programming, i.e. to consider programming-as-leverage, which is reflexive with other domains, meaning that learning the combination of programming and another domain can be easier than learning each separately. According to (Cox, 2005) many of the roots of metacognition in computation are influenced by the large body of studies in cognitive, developmental, and social psychology and the educational and learning sciences.

In this article we integrate the inquiry based learning approach with the computational experiment. Through the methodology of the computational experiment students have to develop models and use the easy java simulator tool to create source code for simple cases, i.e. they are engaged in learning through programming.

In this study, an instructional design model-inquiry based model- was employed for restructuring a prospective teacher education course with the methodology of the computational experiment and the following research question were investigated:

1) To what extent does the Inquiry-Based-Computational Experiment instruction improve prospective primary school teachers’ metacognitive experiences and use of modelling indicators?

2) Are there any correlations between prospective primary school teachers’ metacognitive experiences and learning/cognitive performance?

3) Are there any correlations between prospective primary school teachers’ use of modelling indicators and learning performance?

2. Methodology

Participants
Sixty (N=60) prospective primary school teachers from a Greek University participated voluntarily in the research during the course ‘Didactics of Science Education’. The total number of students was N=79. The course includes the development of algorithms, models and simulation with contemporary pedagogical theories and students have to develop their own pedagogical scenario. Students were studying for their Bachelor degree as primary school teachers at the third year of their studies.

Materials and Procedure
The study, conducted with sixty (N=60) experimental students (prospective primary school teachers) over 13 weeks of an introductory ‘Didactics of Science Education’ course, involved (a) students’ development of a metacognitive strategy relevant for computer modeling, simulation and programming for science, (b) modelling procedures with complete
explanations about strategy use, (c) students’ development of pedagogical scenario in 
alignment with the inquiry based learning and teaching approach.

Self regulatory strategies and knowledge of cognition were assessed through a self-report 
survey administered before and after the treatment using the Metacognitive Awareness 
Inventory tool (MAI) of Schraw and Dennison (1994). Schraw and Denisson (1994) 
developed a 52-item Likert scale self-report inventory for adults (MAI), which measured both 
knowledge of cognition and regulation of cognition. They set out to confirm the existence of 
eight factors, from which three related to knowledge of cognition and five related to 
regulation of cognition. A specimen item on knowledge of cognition is “I use my intellectual 
strengths to compensate for my weaknesses” (conditional part of the cognition of knowledge), 
or “I am a good judge of how well I understand something” (declarative part of the cognition 
of knowledge). The regulation of cognition sub-scale measures knowledge about planning, 
information management strategies, comprehension monitoring and evaluating strategy use. A 
 specimen item on regulation of cognition is “I consider several alternatives to a problem 
before my answer” (comprehension monitoring of regulation of cognition).

Participants had 30 minutes to complete the questionnaire-before and after the 
treatment and rate the statements on a 5-point Likert type–scale 1 (never), 2 (seldom), 
3 (sometimes), 4 (often), 5 (always).

The MAI maximum score (for every subject) is 260 and the minimum is 52.

The first course in Physics is concerned mainly with Mechanics, since Mechanics is an 
esential prerequisite for most of the rest of Physics. Therefore, students’ initial knowledge of 
Mechanics is the most critical to their course performance, so we can restrict our attention to 
that domain of Physics (Halloun, 2006). Students’ prior knowledge and learning outcome 
were assessed using knowledge pre- and post-test, administered before and after exploring the 
Computational Experiment, respectively. To measure the learning performance, the Force 
Concept Inventory (FCI) test (Hestenes, Wells & Swackhamer, 1992) was assessed before the 
treatment (30 questions, maximum score 150, 5 points for each question).

After the intervention, students participated in another test, the Mechanics Baseline test 
(Hestenes & Wells, 1992).

The Mechanics Baseline test (MBT) should be compared with the Force Concept Inventory. 
The Baseline is the next step above the Inventory in mechanics understanding. Questions on 
the FCI were designed to be meaningful to students without formal training in mechanics and 
to elicit their preconceptions about the subject. To the contrary, the Baseline test emphasizes 
concepts that cannot be grasped without formal knowledge about mechanics. The two tests 
are complementary probes for understanding of the most basic Newtonian concepts. Together 
they give a fairly complete profile of this understanding (Hestenes & Wells, 1992). The 
baseline test consists of 26 Questions, 5 points for each question, maximum score 130.

We did not apply the FCI test after the intervention, since our intention was to test students in 
different issues and avoid reproduction of questions.

Both FCI and MAI are standardized tests and their content and face validity has been checked 
(e.g., Hestenes & Wells, 1992; Savinainen & Scott, 2002).

All students participated in the thirteen periods (three hours each) of the course. Initially, 
students were exposed—for six hours-to models in Physics and Mathematics from the 
repository (www.opensourcephysics.org, Retrieved, 21 May, 2013). Models of simulations, 
including in this repository, are developed using the easy java simulator tool (EJS) 
(http://www.um.es/fem/EjsWiki/Main/Download, Retrieved, 13 May, 2013). The Topics were 
selected from the knowledge area of Mechanics.

Easy Java Simulator, also known as EJS or Ejs, is a free authoring tool written in Java that 
helps non-programmers create interactive simulations in Java, mainly for teaching or learning
purposes. EJS has been created by Francisco Esquembre and is part of the Open Source Physics project.

Easy Java Simulations is a software tool (java code generator) designed for the creation of discrete computer simulations.

A discrete computer simulation, or simply a computer simulation, is a computer program that tries to reproduce, for pedagogical or scientific purposes, a natural phenomenon through the visualization of the different states that it can have. Each of these states is described by a set of variables that change in time due to the iteration of a given algorithm.

EJS has been designed to let its user work at a high conceptual level, using a set of simplified tools, and concentrating most of his/her time on the scientific aspects of our simulation, asking the computer to automatically perform all the other necessary but easily automated tasks.

The students were informed of the process they had to follow: a) students write mathematically the model that governs a phenomenon from Mechanics b) students transfer the model to the environment of EJS and construct the algorithm and the ‘view’ panel, c) they decide about the parameters and variables they have to use, and d) they run the simulation from the “view” window of the easy java simulator tool(EJS).

We decided to use the easy java simulator tool for its easiness to be learned, its attractive interface, and its capacity to incorporate all the characteristics of the Java language without advanced knowledge of Java. In addition to that, the “model” window of the EJS environment resembles the usual notebook environment.

The repository www.opensourcephysics.org contains many models that can be explored by the learners. Unlike other Java applet environments, EJS provides the source code for each model, so learners can explore not only the source code but they can also modify the model and see the effect in terms of the model changes. This has the advantage that learners can change not only the variables and the parameters, as it happens in other environments, but they can also change the model (equations, laws etc) itself. Another advantage of this environment is the fact that students realize that the model is somewhat different from the simulation. At the end of the intervention, most students realized that the model is something “static” while the simulation is the “dynamic” part of the process, and that simulation has many forms (event driven, time driven etc).

At the beginning of the intervention, students were asked to complete a questionnaire in order to measure their views about the Modelling indicators as they are described by Hestenes (1999). In doing so, students were exposed to certain models from the repository www.opensourcephysics.org , and they were asked to state their views about the inclusion of Hestenes’s indicators in each model that was presented to them. This process lasted for six hours and students were exposed to four (4) models relevant to Physics Phenomena in Mechanics. For each model, students were asked to explore the inclusion or not of the modelling indicators and to justify their decision.

For example, they had to write what is the system and its surroundings, what are the degrees of freedom, what are the connections/constraints, the interaction laws etc.

After the initial 3 sessions, students were asked to write, as part of their summative evaluation, 2 scenarios that should include the seven features of inquiry and the use of modelling indicators.

Students could develop their own models of simulation from scratch or they could select a model from the repository, www.opensourcephysics.org and modify the model.

To evaluate the use of the modelling indicators, 4 stands for the fully favourable part(systemic structure, geometric structure, temporal structure and interaction structure), 3 for the use of three indicators(less favourable part), 2 for the use of two indicators (partially
favourable part) and 1 for the use of one indicator (unfavourable part). Students were exposed to certain phenomena and were asked to complete the questionnaire for each phenomenon, i.e. if it includes all or a part of the modelling indicators.

Intervention was divided in three phases. For the first nine hours (3 sessions) students were exposed to models from the repository (www.opensourcephysics.org, Retrieved, 21 May, 2013). During these sessions, the Instructor changed the parameters and variables of the models, in some cases the equations of the model, he taught the essential characteristics of the easy java simulator tool, and a dialogue between the instructor and the students was established about the nature of the models, how they are depicted in the computer, what is the method of simulation used, what are the algorithms and how the easy java simulator tool can be used.

After that, the Instructor discussed with the students the seven features of inquiry based process, the spaces of the computational experiment and he developed 3 pedagogical scenario that included models, algorithms, the seven features of inquiry and the modelling indicators. This phase of instruction lasted for 7 sessions.

During this phase students had the chance to be taught about the easy java simulator techniques, how to develop models, and how to combine the three spaces of the computational experiment with the features of inquiry.

For the last three sessions (third phase of the intervention), students presented their drafts in the class as a formative assessment process and received feedback by the Instructor (one of the authors of the study). Feedback included hints about the model, the inclusion of the features of inquiry and the use of the modelling indicators. For each scenario the maximum score was twenty-one (21).

Seven (7) stands for the inclusion of the seven features of inquiry, another 10 points for the model-algorithm development and 4 for the inclusion of the modelling indicators. In the end of the intervention, students were also participated in the MBT test.

Examples of Applications developed

One of the applications developed by the Instructor during the course is the Computational Physics/Computational experiment in the domain of motion under central forces according to the methodology of Molecular Dynamics (Pang, 2006), using the Verlet algorithm, after modifications by the Instructor in the Java code. Source code was written in Java and the Net Beans Toll of Java was used to execute the application.

The computational experiments created were used in order to study the dynamics of the motion of the Comet of mass \( m \) which is confined to move along under the influence of the Gravitational Force

\[
m \mu \frac{d^2 \vec{r}}{dt^2} = -G M \frac{\vec{r}}{r^3}
\]

To deal with the problem, we have to solve Newton’s equations of motion using numerical methods in order to obtain acceleration, velocity, position, energy etc. as functions of time. In Verlet algorithm, we use the central difference for the second derivative:

\[
a_i = \frac{d^2 r_i(t)}{dt^2} = \frac{r_i(t+h) + r_i(t-h) - 2r_i(t)}{h^2}, \text{ while for the velocity}
\]

\[
v_i = \frac{dr_i(t)}{dt} = \frac{r_i(t+h) - r_i(t-h)}{2h}.
\]

The equation of motion is described by Newton’s equation \( m_i \frac{d^2 \vec{r}}{dt^2} = f_i \). For a system of N particles, if we denote

\[M = (m_1, m_2, ..., m_N)\]
\[ F = (f_1, f_2, ..., f_N) \]
\[ R = (r_1, r_2, ..., r_N) \]
and
\[ G = \left( \frac{f_1}{m_1}, \frac{f_2}{m_2}, ..., \frac{f_N}{m_N} \right), \]
we can write Newton’s equations as \( \frac{d^2R}{dt^2} = g \)
\[ \frac{F}{M} = g = \frac{d^2R}{dt^2} = \frac{1}{\tau^2} (R_{k+1} - 2R_k + R_{k-1}) + o(\tau^2) \]

Setting \( t = k\tau, \tau \) (the time step), we get
\[ v_i = \frac{dr_i(t)}{dt} = \frac{r_i(t + h) - r_i(t - h)}{2h}, \]
\[ V_k = \frac{dR}{dt} = \frac{1}{2\tau} (R_{k+1} - R_{k-1}) + O[\tau^2] \]
\[ R_{k+1} = 2R_k - R_{k-1} + \tau^2 \cdot g_k + o(\tau^4) \]

The algorithm used is based on the following aspects
1. Consider the initial values \( r_{\text{max}} = 5.28 \cdot 10^{12} m \) and the corresponding velocity \( v_{\text{min}} = 9.13 \cdot 10^2 m/sec \).
2. For every position of the Comet, we calculate the Comet position relative to the axis \((x, y)\), according to the equations:
\[ x_{k+1} = x_k + \tau V_k + \frac{\tau^2}{2} G_k \]
\[ y_{k+1} = y_k + \tau V_k + \frac{\tau^2}{2} G_k \]
2.1. Calculation of the acceleration according to the equations:
\[ g_{x_{k+1}} = -\kappa x_{k+1} \]
\[ g_{y_{k+1}} = -\kappa y_{k+1} \]
2.2. Calculation of the velocities at \((x, y)\) axis according to the equations:
\[ v_{x_{k+1}} = v_{x_k} + \frac{\tau}{2} \left( g_{x_k} + g_{x_{k+1}} \right) \]
\[ v_{y_{k+1}} = v_{y_k} + \frac{\tau}{2} \left( g_{y_k} + g_{y_{k+1}} \right) \]

The outcome of this Computational Experiment is presented in Figure 2.

Figure 2. Computational Experiment. The interface in Java(EJS) of the motion of the Comet. The blue spot is the Comet and the yellow one the Sun.

One of students’ simulations is presented in Figure 3.

Figure 3. An example of students’ application using EJS.

For every session of the 3-hours course, students were engaged in scientifically oriented questions prompted mainly by the teacher. At each session, students examined the model, they were asked to express their views on the different spaces of the CE, the variables and parameters to include in the spaces and to create their own models taking into account the seven features of the inquiry process.
3. Results

For dataset smaller than 2000 elements, Shapiro-Wilk test, otherwise, the Kolmogorov-Smirnov test is used. In our case, since we have only 60 elements, the Shapiro-Wilk test was used.

3.1 Descriptive Statistics

In Table 2 we present the results for the variables Learning Performance, Metacognition and Modelling Indicators before the Intervention.

Table 2. Descriptive Statistics for the quantitative variables of the study, before the intervention

In Table 3 we present the results for the variables Learning Performance, Metacognition and Modelling Indicators after the Intervention.

Table 3. Descriptive Statistics for the quantitative variables of the study, after the intervention

We also present the histograms for some variables from which we verify the use of parametric methods. All the variables satisfied the Shapiro-Wilk test (Kolmogorov-Smirnoff test was also satisfied) for the normality of their distribution.

In Figure 4 we present the histogram for the Metacognitive Experiences Before the Intervention. The value of the Shapiro-Wilk test is 0,158 and that of Kolmogorov-Smirnoff is 0,2.

Figure 4. Normal Distribution of the Variable Metacognitive Experiences Before the Intervention

The distributions for the other variables are also normal and we do not present them.

The aim of this study is to explore whether the use of the CE methodology has an impact on learning performance, metacognitive experiences, use of modelling indicators and model and algorithm development in the form of a pedagogical scenario.

For this reason, we performed t-tests for all the quantitative variables before and after the intervention. The mean values for the FCI and MBT tests differ significantly (t=28.36, df=59, p<0.001).

There is also a significant difference in the mean values for the metacognitive experiences before and after the intervention (t=6.84, df=59, p<0.001).

Modelling indicators differ also (t=10.427, df=59, p<0.001).

Using stepwise regression analysis, with the variable “use of the inquiry features” as independent variable, we notice that 93% of total variance can be interpreted from the variable “model and algorithm development” (F1.58=778.892, p<0.001), while, if we increase the variable “model and algorithm development” by one unit, “use of the inquiry features” will be increased by 0.965.
The correlation of these variables is $r=0.965$.

Using stepwise regression analysis, with the variable “MBT Test results” as independent variable, we notice that 82% of total variance can be interpreted from the variable “model and algorithm development” ($F_{1.58}=268.558$, $p<0.001$), while, if we increase the variable “model and algorithm development” by one unit, “use of the inquiry features” will be increased by 0.91.

Using stepwise regression analysis, with the variable “Metacognitive Experiences after the intervention” as independent variable, we notice that 75% of total variance can be interpreted from the variable “model and algorithm development” ($F_{1.58}=179.177$, $p<0.001$), while, if we increase the variable “model and algorithm development” by one unit, “use of the inquiry features” will be increased by 0.86.

4. Conclusions and Discussion

Metacognitive knowledge includes knowledge about oneself as a learner and the factors that might impact performance, knowledge about strategies, and knowledge about when and why to use strategies of thinking (Lai, 2011; Pearson’s research report series, http://www.pearsonassessments.com/hai/images/tmrs/metacognition_Literature_Review_Final.pdf, Retrieved, 15 May 2013). Metacognitive regulation is the monitoring of one’s cognition and includes planning activities, awareness of comprehension and task performance, and evaluation of the efficacy of monitoring processes and strategies. Metacognition can be improved with appropriate instruction, with empirical evidence supporting the notion that students can be taught to reflect on their own thinking (Lai, 2011; Pearson’s research report series, http://www.pearsonassessments.com/hai/images/tmrs/metacognition_Literature_Review_Final.pdf, Retrieved, 15 May 2013). As Kuhn and Dean (2004) explain, metacognition is what enables a student, who has been taught a particular strategy in a particular problem context to retrieve and deploy that strategy in a similar but new context. Our methodology was to teach students particular strategies using pedagogical scenarios, and next let students to deploy their own strategies during the development of their scenario. In this article, we support the Computational Experiment (CE) methodology as a key challenge for the development of interactive-inquiry learning environment and metacognitive experience increase, where students are stimulated to develop modelling skills, be engaged in inquiry tasks and develop strategies to solve particular problems in order to enhance their metacognitive awareness and evaluate their efficacy of strategies. In the Computational Experiment approach, strategies consist mainly of the model creation and the algorithm generated during the problem solving and this justifies the strong impact of the variable “model and algorithm development” on metacognitive experiences, use of inquiry features and MBT test in Physics. Our approach was based on the issue that the computational experiment (CE) can serve as methodology and as modelling tool for developing explicit models of problems that students are trying to solve during their involvement in inquiry processes. That is, rather than modelling the domain knowledge from which problems are extracted, the computational experiment approach favours the modelling of factors and the entities on a problem- basis process. Our results indicate that an essential factor for the implementation of ICT in Education is the use of models and algorithms and the understanding of modelling indicators in the framework of the computational experiment. In all models exposed to students, the Instructor tried to include the essential features of Inquiry Based approach and to combine inquiry activities with the
three spaces of the Computational approach. In most of the cases we used the classification of Table 1 and the process described in Figure 1. Students seized this methodology and they considered that as a standard approach to develop their activities. Initially students identified Inquiry as a collection and explanation of evidence, but, in due course, they connected Inquiry with engagement in scientifically-oriented questions, the development of scientifically proper hypotheses and the comparison of their own explanations to existing scientific explanations and models. During discussions with students, we noticed that while initially, students selected the hypotheses and the variables to be involved, after involvement in the modelling process, they wanted to return back to the hypotheses space of CE and update the variables and the relations between them. At the experimental space, students executed their model and algorithm and collected their data. At this phase they also corrected the bugs and compared their data with those from textbooks. During the development of their models, students determined what constitutes evidence and analysis of evidence, while they used their data to form explanations. We observed that the Explain feature of inquiry was –mainly– formed by the students, as well as the Evidence feature, while for the feature of Analysis students asked help from the Instructor. Students formed links to explanations mainly by comparison with explanations from textbooks but they considered that data were not always in align with the theoretical principles and they started to think about the types of errors(systematic, random) that usually are not discussed in detail in the textbooks. Regarding the Inquiry Feature “Communicate”– it was mainly coached by the Instructor, who provided scaffolding guidelines and procedures for communication and justification of explanations. While the relation of model development with the MBT results could be considered as an expected outcome, the relation of metacognitive experiences to the “model and algorithm development” is a very important result of the research and it should be further analyses in other research studies. Most researchers agree that metacognition is an important construct to study, but difficult to measure and central to the problems relating to metacognition is finding ways to favour the development of metacognitive skills in inquiry environments. Our results show that students, who are skilled in model and algorithm development are more strategically thinking and perform better than those who are unaware of working on their own mental system (Schraw and Dennison, 1994; Schraw, 2000). During the learning and teaching sequences based on CE-Inquiry environments, students considered that they followed a concrete strategy and methodology which allowed them to become more metacognitive aware learners, more strategic and this caused enhancement of their ability to plan, sequence, and monitor their learning in a way that directly improves performance. Flavell (1976) argues that metacognitive experiences that allow one to monitor and regulate one’s cognition play a major role in the development and refinement of metacognitive knowledge. Computational experiment serves the tools to create a learning environment that is more learner-centred by providing a greater variety of resources for development of models and selection of simulation techniques, which allows students to follow their own learning pathways enhancing learners’ regulation of cognition and enhance their metacognitive experiences. Results support the view that use of the computational experiment approach, advance students’ metacognitive experiences, since the computational experiment offers the chance to be engaged in reflection processes and regulation of cognition at the different stages of the experiment, following the three spaces of the CE. Finally, by being
engaged in representing scientific concepts, relationships of variables, algorithm development and creation of models of simulation with the use of EJS, students changed their disposition towards the nature of science and they were more critical to scientific issues and results. They also advanced their inquiry skills in alignment with other research results (Cuevas et al., 2005). Our results show that CE-Inquiry Based environments can trigger off students’ motivation to understand fundamental issues and to develop critical and inquiry skills as well as modelling skills in alignment with the results by (Squire et al., 2004; Bell et al, 2010).

References


Histogram

- Mean = 139.87
- Std. Dev. = 50.114
- N = 90

Frequency

Metacognitive Experiences Before
Highlights

- Models of simulations based on algorithms were developed
- We examined the impact of the computational experiment on metacognitive experiences
- Learning performance is strongly affected by the computational experiment methodology
- There was a significant shift in metacognitive experiences
- There was a significant shift in use of modelling indicators