Cognitive Aspects of Computational Modeling and Simulation in Teaching and Learning

Osman Yaşar
The College at Brockport
State University of New York
Brockport, NY 14420
Tel: +1 (585) 395-2595
oyasar@brockport.edu

ABSTRACT

We discuss cognitive aspects of modeling and simulation in an efficacy study of computational pedagogical content knowledge professional development of K-12 STEM teachers. Evidence includes data from a wide range of educational settings over the past ten years. We present a computational model of the mind based on an iterative cycle of deductive and inductive cognitive processes. The model is aligned with empirical research from cognitive psychology and neuroscience and it opens door to a whole series of future studies on computational thinking.

General Terms

Computational Theory of Mind, K-12 Teaching and Learning

Keywords

Deductive and Inductive, Cognitive Processes, Memory Retrieval

1. INTRODUCTION

Educators structure training and curriculum based on learning theories of how the human mind works. Recent findings from empirical research by cognitive psychologists and neuroscientists have created a critical mass to change the way we prepare teachers and support their classroom instruction. This is an opportune time for computer science educators to ground in cognitive theories the well-known concepts and processes in computational science.

Make it Stick, an ostensibly groundbreaking book published recently and coauthored by several prominent cognitive scientists has turned conventional ideas of learning upside down (Brown *et al*. 2014). The book offers many sound practices to help students easily retrieve content they learned in class, retain it, and apply it in different contexts to solve problems. New research suggests that repeated, delayed and interleaved retrievals make new concepts stick in memory longer if the process is effortful (pp. 47). Learning is mediated by memory, because human brain attempts to interpret new concepts in terms of previously registered knowledge and facts. However, some degree of forgetting is also good for learning because it forces the learner to use effort to cognitively engage oneself to recall or reconstruct newly acquired concepts through different neural pathways or links that exists and are retrievable.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

According to neuroscience, information is stored into the memory in the form of a specific pattern of neurons placed on a pathway and fired together (Restak 2001, Brown *et al.* 2014). The number and strength of such pathways improve the storage and retrieval of information. A memory or a newly learned concept can be a combination of previously formed memories, each of which might also involve a vast network of concepts and details mapped onto the brain's neural network in a hierarchical way shown in Fig. 1.

The key to storing a concept more permanently into the memory is to link it to previously stored basic and retrievable concepts. And, the more links to associated concepts, the higher the chances of recalling this concept when needed later. Spaced-out cognitive retrieval practices attempted at different times, various settings and contexts is good because every time the recall is attempted it establishes more links that will help the remembering and learning. Exposure to new concepts through links to multiple views from different fields of study is, therefore, an effective retrieval strategy recommended by cognitive psychologists (Brown *et al.* 2014). This is called *interleaved retrieval* practice and it now forms a cognitive foundation for the computational pedagogical content knowledge (CPACK) framework that we developed for teacher professional development (Yaşar *et al.* 2015). In the following Sections (2.1 - 2.5) we describe theoretical foundation of CPACK followed by its implementation and impact on teaching and learning (Sec. 3) in secondary school classrooms.

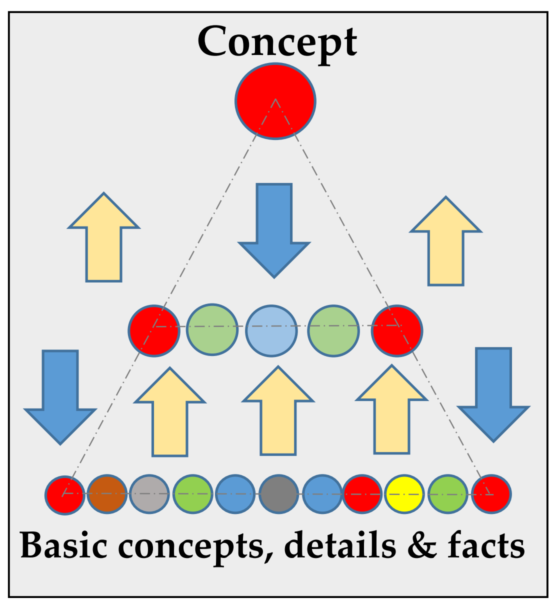

Figure. 1: Distributive and associative aspects of information storage and processing (Yaşar 2015).

2. THEORETICAL FOUNDATION

2.1 Interdisciplinary Education

Interleaving retrieval practices by weaving together multi-disciplinary features around a common topic (i.e., interdisciplinary education) has great advantages for gaining deep and lasting knowledge but it is not easy for several reasons. It would require a more cognitive effort than usual and as such, it would slow down the process of learning. In college, it would delay graduation and in public schools' packed schedules it would risk compliance with local and state-mandated curriculum. Technology can be used to speed up this interdisciplinary learning but it needs training of teachers to teach content in pedagogically appropriate ways, thereby requiring a close integration of technology, pedagogy, and content as shown in Fig. 2. Recently, a theoretical framework, namely technological pedagogical content knowledge (TPACK), has been developed by Mishra & Koehler

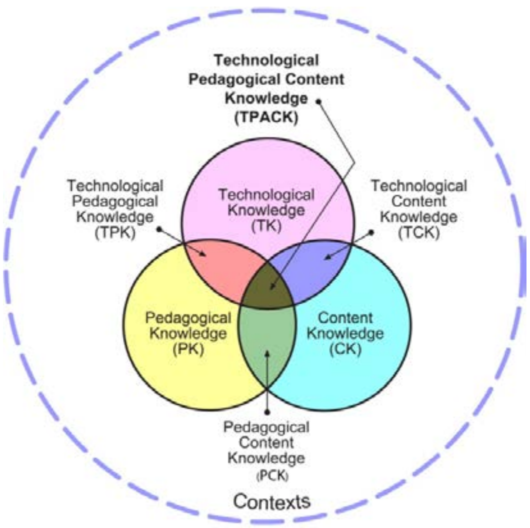

Figure 2: TPACK framework (Mishra & Koehler 2006).

(2006) to address challenges of T, P, and C integration. Practicing teachers have been offered professional development (PD) to help them deploy appropriate technologies in the classroom, stay up-to-date with emerging technologies, and assess efficacies of different pedagogical approaches (Loucks-Horsley *et al.* 2010). But, due to frequent changes in available tools, challenges might never go away as far as transferring curriculum inventories and PD content to new circumstances. Furthermore, teaching with technology often requires customization and the needed technologies must be both content specific and pedagogically suitable at the same time (Koehler & Mishra 2008). While the latest technologies offer more capacity for applicability, their optimum utilization may necessitate knowledge of tools' operational underlying principles for easier transfer into new circumstances and better integration (Koehler & Mishra 2008, Niess 2005, Flick & Bell 2000).

It is not very common to come across presentations or papers in teacher education conferences that report use of a pedagogically appropriate technology that is widely applicable to topics in a STEM content area. It is even less uncommon to see one that applies to teaching of topics in multiple content areas. This is what led scientists such as us who heavily used computational modeling and simulation technology (C-MST) in scientific research in the past several decades to cross paths with pedagogy and teacher education experts. We need their help to get more and better students from public schools to enter computational science programs and they need help with interdisciplinary TPACK training of teachers. At the 2014 and 2015 SITE (Society for Information Technology and Teacher Education) conferences, we presented a case study (i.e., CPACK) by demonstrating how we have integrated computational methodology and technology into teacher education. Encouraged by a warm reception and a TPACK paper award (Yaşar *et al.* 2015) from the SITE education community, we started a fruitful collaboration with other researchers and this has resulted in a better understanding of cognitive foundations of computational modeling and simulations.

There is an important feature of interdisciplinary education that can be best described by Aristotle's well-known statement, "the whole is more than the sum of its parts," or the theory of Gestalt psychology, "the whole is other than the sum of its parts," which means that the whole has a reality of its own, independent of the parts (Koffka 1935). Accordingly, educators have noted an emerging nature of TPACK when technology, pedagogy, and content closely interact (Mishra & Koehler 2006), which is illustrated as the overlap of Venn diagrams in Fig. 2. There is even a stronger case, CPACK, when mathematics, computing, and sciences are integrated through CMST (see Fig. 3). Not only has it given rise to a new content domain of computational science as witnessed by degree programs in the past two decades (Swanson 2002, Little 2003, Yaşar & Landau 2003) but it also led to a particular pedagogy which was not even there among the constitutive domains of computing, mathematics, and sciences to start with (Yaşar & Maliekal 2014a). Below, we explain cognitive foundations of this computational pedagogy.

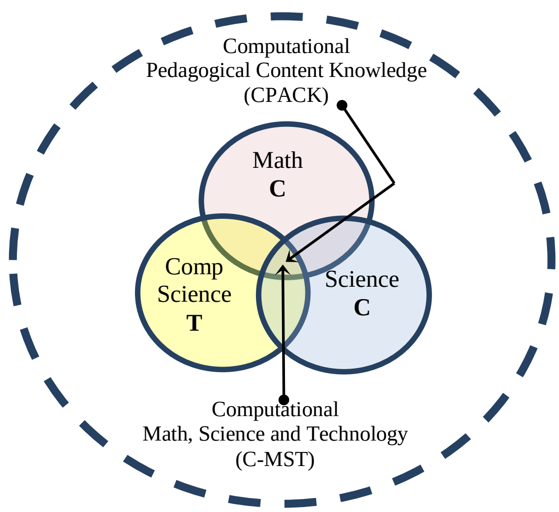

Figure 3: CPACK framework. While pedagogy is a separate domain in TPACK, it shows up inherently here as an outcome of interdependencies of computing, math, science and technology.

2.2 Mind as a Computational Device

Modeling and testing has been an important tool for scientific and engineering research for hundreds of years. Scientists often start with a model (e.g., a hypothesis or a concept) deductively based on the current research, facts, and information. They test the model's predictions against experimental data. If results do not match, they, then break down the model into its parts (sub models) to identify what needs to be tweaked. They retest the revised model through what-if scenarios by changing relevant parameters and characteristics of the sub models. By putting together new findings and relationships inductively among sub models, the initial model gets revised again. This (deductive/inductive) cycle of modeling, testing, what-if scenarios, synthesis, decision-making, and re-modeling is repeated — similar to the bidirectional distributive/associative structure in Fig. 1 — as resources permit until there is confidence in the revised model's validity.

In recent years, computers have been very effective in conducting scientific research because they speed up the model building and testing of different scenarios through simulations that provide quick feedback to researchers in order to improve the initial model (NSF Blue Ribbon Report 2006). CMST's role in scientific and industrial research was proven beyond doubt when computational predictions matched behavior of physical models in high-stake cases (e.g., safety of cars and planes, emissions from engines, and approaching storms). Its use was uniquely justified when a study was impossible to do experimentally because of its size (too big such as the universe or too small such as subatomic systems), environmental conditions (too hot or dangerous) or cost. CMST eventually demonstrated to be generating innovation and insight, just like experimental and theoretical research and this ultimately led to the recognition of computation by the scientific community as a third pillar of doing science besides theory and experiment (PITAC Report 2005).

While such capacity was available only to a small group of scientists in national labs, their demand for computationally competent post-docs and doctoral students led to graduate programs in research universities. A dramatic increase in access to and power of high performance computing and the drop in its cost in the past 20 years helped spread the use of CMST tools into the manufacturing industry. Driven by market needs and trends, rather than empirical research into their effectiveness in education, funding agencies and colleges started investing in new CMST-based BS and MS degree programs across the world (Swanson 2002; SIAM Report 2001, 2007, Yaşar *et al*. 2000). It was not until friendly versions of such tools were available and considered for use in K-12 settings that a detailed and thorough empirical research was undertaken to measure their effectiveness in education.

If used appropriately, CMST tools can involve students in inquiry-based, authentic science practices that are highlighted in the recent framework for K-12 science education (NRC 2012). A growing body of research (Bell & Smetana 2008; Wieman *et al.* 2008) identifies computer simulation as an exemplar of inquiry-guided (inductive) learning through students' active and increasingly independent investigation of questions, problems and issues. Research into the use of computer simulations in science education has been reviewed periodically and quite frequently in recent years. These include early efforts by de Jong & van Joolingen (1998) and by Bell & Smetana (2008), as well as recent efforts by Rutten *et al.* (2012) and by Smetana & Bell (2012). The article by the Rutten *et al.* (2012) reviewed (quasi) experimental research in the past decade (2001-2010) and the one by Smetana & Bell (2012) reviewed outcomes of 61 empirical studies since 1972. The overall findings support effectiveness of computer simulations. In many ways simulation has been found to be even more effective than traditional instructional practices. In particular, the literature shows that computer simulations can be effective in: 1) developing science content knowledge and process skills, and 2) promoting inquiry-based learning and conceptual change. Effectiveness of CMST in education is also well grounded in contemporary learning theories that recognize the role of experience, abstract thinking, and reflection in constructing knowledge and developing ideas and skills (Hammond 2001; Donovan & Bransford 2005; Illeris 2009; Mooney 2013).

Since CMST is beneficial to both scientists and students in their inquiry and learning, one might wonder in what ways it resonates with the basic functions of the mind. Although the literature suggests linking modeling and simulation to some cognitive functions such as abstract thinking and decomposition skills (Wing 2006), empirical research in cognitive psychology and neuroscience (Brown *et al*. 2014) encourages us to search further, as there might be a deeper link at more fundamental levels. For example, according to the computational theory of mind (CTOM), the deepest link between electronic and biological (mental) computing devices is a) the common nature of the information that they both process, and b) the way that they process it (i.e., addition & subtraction), regardless of the underlying infrastructure that does the computation (Montague 2006).

Many fields have their hands in the study of how learning takes place in the mind. Cognitive psychologists try to understand how the mind works through empirical research into how people perceive, remember, and think. Developmental and educational psychologists form theories of human development and how they can be used in education. At the same time, neuroscientists use imaging techniques to understand the brain mechanisms that take part in learning. What was started by Alan Turing, the father of computer science, still continues to shed light today on the study of the mind. Basically, Turing's idea was that if thoughts (i.e., information) can be broken up into simple algorithmic steps, then, machines can add, subtract or rearrange them as our brains do (Montague 2006; pp. 6). Turing also provided an insight that there should be a distinction between the patterns of computations (e.g., computer software and mind) running on a device and the device parts (e.g., computer hardware and brain). His insight keeps fueling the work of computer, computational, and cognitive scientists (Montague 2006; pp. 7). Basically, he laid foundations of a devise that could imitate the mind, thereby giving us a simplified representation (model) of the mind to understand how it would work in different contexts.

While CTOM played a central role within the cognitive sciences during 1960s and 1970s, modern philosophers think that equating mental representations with information processing leaves out the meaning associated with mental events (Montague 2006; pp.8). We know that CTOM is far from complete, as information processing alone cannot define mental states. But, we also know from scientific research that computational modeling and simulation can generate insight when done in a bi-directional iterative way as shown in Fig. 1. If today's advanced computer hardware and software have grown to a capacity to generate insight and conceptual change through a structured and cyclic computation with many levels involving various sizes and constructs of information at each level, then we should investigate if the same structure and mechanism support fundamental cognitive processes that may be common to both biological and electronic computation.

In his book, "*How We Make Decisions*," the neuroscientist Montague (2006), an ardent supporter of CTOM, describes how the mind attaches value to the computations in order to make meaningful decisions. He argues that the concern for survival pressures us to be efficient in the way we consume our available energy. As an extremely efficient computational device, the brain actually runs on orders of magnitude less electricity than mechanistic computers and mobile devices (p. 26). Furthermore, he suggests that the concern for efficiency makes us assign "value" to our thoughts, decisions and actions by computing and evaluating different scenarios before we take an action (p. 51). And, that, he thinks is the root of our intelligence and why we have pushed ourselves to be smarter over time.

2.3 Electronic & Biological Computation

Humans have long been curious about how the mind works in ways that are meaningful, plausible, and fruitful for further research possibilities. Studying the mind has been much complicated as it takes place in a delicate, inaccessible, and complicated organ, the brain. However, consideration of the information in terms of simpler and computable pieces by Alan Turing led to an electronic device to imitate the biological brain. After almost a century, the imitation has gotten so complicated, both structurally and functionally, that we may be able to discover how the original (mind) computes by studying how the imitation (computer) does it. Yet, despite similarities of computational processes between electronic and biological computing devices, each uses a different hardware to accomplish what it does. While electronic computers have evolved into distributed structures like the brain's neural network, there exist many differences. Much of the literature on "computation" today refers to how it is done on electronic devices and it may be time to use the term computation in a device-independent way.

As briefly mentioned in the introduction, the latest neuroscience studies now shed light on how information storage, retrieval (remembering), and processing (thinking) take place by the brain hardware (Brown *et al.* 2014). While electronic computing machines handle information storage and processing separately through different hardware components, our brains have no separate place for information storage — storing and retrieval are part of information processing (thinking). Both the long-term storage and processing of information involve a synchronized distributed participation of all neurons in related regions of the brain (MacDonald 2008: 97). Programmers of parallel computers know that management and utilization of a distributed hardware necessitates *scatter* and *gather* type communication functionalities in software. That is similar to what is going on in the brain circuitry. When new information arrives, it lights up all related cues, neurons and pathways in a *distributive* process that is similar to the top-down action in Fig. 1, where new concept is broken up into related pieces. With the same token, retrieving a memory is a reassembly of its original pattern of neurons and pathways in an *associative* process that is similar to the bottom-up action in Fig. 1. Retrieval is often regarded as an act of creative re-imagination and what is retrieved is probably not the original pattern but one with some holes or extra bits (Brown *et al.* 2014: 75, MacDonald 2008: 101). Neuroscientists argue now that there is no distinction between the act of remembering and thinking (MacDonald 2008: 97).

The distributive and associative way of information processing by the brain circuitry is consistent with the dual deductive and inductive process of computational modeling and simulation that we discussed in earlier sections. While the brain's neural circuitry offers a chance for full utilization, the efficiency, intactness, and effort-fullness with which it is used depends on each individual. A scientist is a good example of a person who exercises this bi-directional thinking methodology in a complete cycle. Since the latest learning theories recommend that student learn science the way a scientist does his inquiries, these thinking skills should then be taught to young learners. They are actually part of the electronic computational thinking (CT) skill set as described by Jeannette Wing (2006). Some of the currently described CT skills may be grounded in cognitive processes that we have discussed here. For example, the decomposition skills of CT roughly correspond to the distributive, deductive, and top-down cognitive process of information we have described here. And, the abstraction skills roughly correspond to what we have described as associative, inductive, and bottom-up cognitive process of information.

Abstraction is an *inductive* process, whereby details are filtered out and focus is placed on more general patterns, thereby allowing one to assign priority and importance to the newly acquired information. Researchers find it amazing that we make strong generalizations from sparse, noisy, and ambiguous data (Tenenbaum *et al.* 2011). Abstraction helps our cognition, especially at its developmental stages, by simplifying, categorizing, and registering key information and knowledge for quicker retrieval and processing (Bransford et al. 2000). Perhaps, we developed abstract thinking skills as a result of a survival concern for having limited resources (*i.e., time, memory, attention*). Our tendency to summarize and generalize information ─ before we permanently store it ─ might be a strategy to overcome limited storage capacity. Such tendency can shield us from details that have no practical value for survival. Another evolutionary idea is that the brain's tendency to process information in a dual fashion might be because it has sought a way to adjust to dual behavior of matter and the incoming information that reflects matter's dual behavior. Whatever the origins are, findings in neuroscience indicate that it is not just the limited capacity of our brain or our survival instinct but also the distributed structure of the brain hardware that drives a bi-directional (distributive and associative) flow of information, which results in tendencies that benefit us.

The growth of our brain hardware and software is a bit complex and many things can go wrong during a lifespan. Normally, at birth, the circuitry at the inner part of the brain is up and running to manage vital and involuntary functions (e.g., breathing, heartbeat, and some degree of sound and visual tracking), but the outer part (cerebral cortex) takes some time to be ready for voluntary actions (e.g., conscious thought, information storage and processing) (Restak 2001). Actually, the majority of neurons that a human is born with are contained within this thin cortex that separates humans from other animals. While only a few neurons develop during adulthood, we can take comfort that mental growth is not solely based on the number of neurons in the brain, but rather the increasing complexity of the connections between them. Other factors that affect mental growth include the functionality that each neuron or groups of neurons assume, the size they grow into, and the placement in different parts of the brain that they migrate towards. Even more important is the number of inter-neuronal connections, which are estimated to be near 100 trillion. New neural connections are being made all the time as we learn new things. In fact, these connections constitute the definition of learning, and the existing connections are strengthened, weakened, or even eliminated if not revisited often enough. Genetics plays only a partial role determining the growth of the brain, as there are not enough genes on the human chromosome to code for the placement of billions of neurons and trillions of connections (Restak 2001). This luckily leaves plenty of room for the brain (and the mind) to continue growing as a result of one's free will, experience, and environment.

So, the good news is both deductive (e.g., decomposition) and inductive (e.g., abstraction) thinking skills can be improved beyond what is inherited, through training, education, additional knowledge and experience. In computer science, we use abstraction skills heavily and students get opportunities to sharpen them while writing large-scale complex codes (such as operating systems, compilers, and networking) in which the complexity is distributed into seemingly independent layers and protocols of the code in such a way to hide the details of how each layer does the requested service (Armoni 2013). Decomposition skills are also equally important in computational and mathematical problem solving. When facing a complicated situation (just like a complex science concept), one is often advised to divide (scatter) the complexity into smaller pieces and then attack each one separately until a cumulative (gather) solution is found. For example, domain decomposition is a common method in parallel computing to distribute the workload among multiple processors. In mathematics and physics, the Fourier series offers great benefits to deal with seemingly complex periodic functions by decomposing them into the sum of a set of simpler, namely *sines* and *cosines*, functions. In public culture, the famous “divide and conquer” phrase, supposedly by Napoleon, as well as ‘many a little makes a mickle’ by Benjamin Franklin all point to our awareness of the importance of the decomposition strategy. But, as stated above, not everyone is equally aware of the importance of such skills, nor are we all practicing and utilizing them fully and equally. So, some of us educate others, and in doing so, we have historically chosen different methods, as explained below, based on circumstances and needs. The good news is that technology (e.g., CMST) has now made it possible to combine seemingly competing and disparate methods into one that might do it all.

2.4 Learning Processes Supported by CMST

The issue of why STEM subjects may not be as engaging as others is complex. According to a study in 20 developed countries (Sjøberg & Schreiner 2005, Osborne & Dillon 2008), student attitudes towards science become increasingly negative as a country advances economically. The study suggests this phenomenon to be deeply cultural. Born in the early-to-mid 20th century as a reaction to the rigid and formal style of discipline-based education, today’s progressive education system in the U.S. continues to engage students by making learning fun and exciting (Mooney 2013). There is nothing wrong with that. However, learning some subjects, such as science and mathematics, can be overwhelming because it involves factual details and requires application, discipline and delayed gratification ─ values the contemporary culture does not seem to encourage. Effortful learning is the key as we discussed earlier, according to the latest research in cognitive sciences and neuroscience. While the need for guiding young minds into the process of effortful learning had already been theorized by Vygotsky around the time of progressive education movement in America, the theory did not find its way across the Atlantic until two decades ago (Mooney 2013; Hammond *et al.* 2001).

There is no doubt that factual details in science and mathematics coursework are often overwhelming, causing high degrees of frustration for some students. Such individuals perceive science and mathematics topics to be more complex than they are and abandon their pursuit altogether. However, learning can be a joyful activity, if one is predisposed to delayed gratification, which is seldom the case with middle and high schoolers. Hence teachers everywhere face challenges that are daunting. Perhaps, there are two ways to overcome this. One of them requires a cultural change to teach new generations how to become effortful learners and predispose them to delayed gratification. This would take a whole village to do. And, it might take a lot longer than we have come to know Vygotsky’s theory, which says pushing a learner to reach his potential is a lot more important than giving him freedom to choose between effort and withdrawal. This would be like swimming against the flow in today’s educational system and cultural setting. The other option requires a pedagogical practice to employ a general simplistic framework from which instructors can introduce a topic and then move deeper with more content only after students gain a level of interest to help them endure the hardships. As explained in the next section, educators have often opted for this latter *deductive* approach.

Teacher organizations and national standards (Bell *et al.* 2008) have suggested ways to create “antidotes” from the very thing (technology) that is known to have caused distraction and a tendency for an easy living. At the same time, the latest learning theories suggest that students should learn science the way scientists do their work (Bransford *et al.* 2000). For example, the framework for next generation science standards (NRC 2012) suggests that students learn better if they are engaged in activities closely resembling the way scientists think and work. If we combine these suggestions ─ that is, using technology with the way scientists conduct their work ─ we would recall from Section 2.2 that scientists today heavily use CMST to do their work. So, the antidote can be computational modeling and simulation but it has some strings attached to it according to a national report (NSF Report 2008). Young learners cannot use the same CMST tools that the scientists use, as they might need prerequisite knowledge that they surely will not have. The report states that at early stages computational modeling approach should involve *easy experimentation* (learners must be able to quickly set up and run a model using an intuitive user interface, with no knowledge of programming or system commands) and *high interactivity* (models need to evolve quickly and include smooth visualizations for providing interactions and feedback to users).

Modeling is a simplification of reality ─ it eliminates the details and draws attention to what is being studied. It enables the learner to grasp important facts surrounding a topic before revealing the underlying details. Tools, such as those in Table 1, now make it possible for instructors to offer easy experimentation in the classroom without having to expose students to STEM principles. For example, as described in later sections, Interactive Physics (IP) and AgentSheets (AS) can be used to create many fun things that could engage students into science experimentation, either by modifying an existing model or creating one from scratch.

Table 1. List of CMST tools used in the CPACK PD.

Interactive Physics (IP): investigate concepts in physics without prior physics background. http://www.design-simulation.com/IP.
AgentSheets (AS): create games and simulations through agents and rules of engagement. http://www.agentsheets.com.
STELLA: model a system by a pictorial diagram of initial values and rate of change equations. http://www.iseesystems.com.
Geometer’s Sketchpad (GSP): model geometrical concepts; compute distances, angles & areas. http://www.dynamicgeometry.com.
Project Interactivate (PI): online courseware for exploring scientific and mathematical concepts. http://www.shodor.org.
Excel Spreadsheets: conduct modeling and simulations using a simple algebraic (new = old + change) for rate of change.
Texas Instruments (TI) Tools: advanced graphing tools to conduct algebra, functions, and rates of change

Simulation adds another level of benefit on top of easy modeling by providing a dynamic medium for the learner to conduct scientific experiments in a friendly, playful, predictive, eventful, and interactive way to test hypothetical scenarios. For example, in a harmonic motion of an object attached to a spring (Fig. 4), IP can provide control buttons to change physical parameters such as string constant, mass of the swinging object and its initial velocity, intensity of gravitational acceleration, among others. It also gives the user the ability to change some operational parameters, such as the run-time and accuracy desired from the simulation. Furthermore, it allows the learner to go into the initial model's details and break it into its constitutive parts in order to run various *what-if* scenarios. Based on these scenarios and their outcomes, the learner can go back to the design phase and change the model (spring and box) to his desire. This dynamics of making decisions that lead to modifications to the initial model based on what-if scenarios is an *inductive process* because it lets the learner to put pieces of the puzzle to come up with a revised model. When used together, then, modeling and simulations affords the learner the opportunity to cycle iteratively back and forth between the inductive and deductive approaches to learning (Yaşar & Maliekal 2014). This resonates with how the mind itself works because it, too, uses a similar dual methodology (distributive and associative) in its information storage and processing as we explained before.

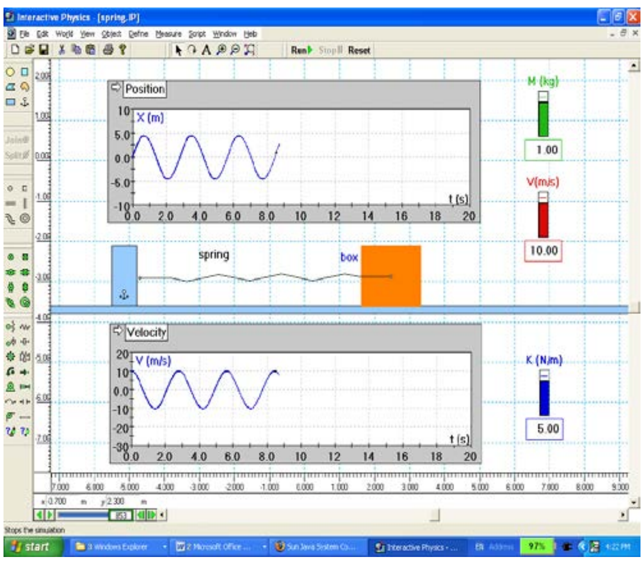

Figure 4. A typical user-created simulation in Interactive Physics: harmonic motion of a box attached to a spring on a flat surface.

2.5 Deductive & Inductive Approach to Instruction

There are many advantages of deductive and inductive approach in teaching and learning. The deductive approach to instruction entails the teacher introducing a new concept or theory to students by explaining it first, then showing an application or two of the theory or concept, and wrapping up the instruction by affording students an opportunity to apply the theory or concept by completing homework problems (Prince & Felder 2006). This has been and continues to be the traditional approach to science instruction, and it often leads to apathy and eventual attrition of students. The inductive approach to instruction, by contrast, first presents students with a problem, a case, or data from an experiment. Students are then guided to explore underlying facts, issues and the like. As the culminating step, students are led to acquire on their own an understanding of the underlying concept or organizing principle (Prince & Felder 2007). Inquiry-guided learning, problem-based learning, and project-based learning are all among forms of inductive instruction. While empirical evidence suggests that the inductive approach to instruction is superior and that it fosters greater intellectual growth (Bransford et al. 2000, Donovan & Bransford 2005), prudent educators should take advantage of different approaches of teaching.

Modeling- and simulation-based computational pedagogy carries many characteristics of the *constructivist* approach (Grabinger & Dunlap 1995), including inquiry-based, generative, cooperative, and interactive learning as well as project and team based instruction. Creating a model through step-wise process and running it at each stage of the development have the added advantage that learners get immediate feedback about their work. It may be used in situations when learning about the underlying theories and mathematical concepts that are important. Through this process, learners can be led to develop an understanding of scientific reductionism that studying a system or solving a complex problem requires breaking the system into its components or the complex problem into smaller chunks (i.e., decomposition). Using models and simulations, learners become actively engaged in "doing," rather than passively "receiving" knowledge. In so doing, the learner becomes the center of the learning process, allowing self-interpretation of the problem and revise it if necessary, mediated by own biases, beliefs, preconceptions, prior knowledge and observations. Once learners successfully infer an organizing principle or theory, they can embark on the next logical and necessary step; one that involves predicting the consequences of the organizing principle or theory that learner just inferred and ascertaining whether the organizing principle or theory is viable, given the consequences. Anyone who learns in this fashion would, in fact, be practicing the craft of scientists (Wieman *et al.* 2008).

Because simulation modules of differing complexity and flexibility have already been developed and made public, it is now possible to lead learners to perform a series of simulations to explore a scientific process in a manner that is similar to how scientists conduct controlled experiments, by holding all except one variable invariant. A teaching and learning method reliant on CMST is being welcomed by today's traditional college and school students, as they are digital natives, attracted to and captivated by all things digital! Even non-science students, with no prior knowledge of physics, who used CMST tools and web-based simulations, have shown the ability to provide good explanations of scientific phenomena much more quickly (within hours) than physics majors after a year of physics (Wieman *et al.* 2008). So, having believed in the promise of dual pedagogical aspects of CMST, we ran a professional development program for in-service and pre-service teachers, hoping that it would engage teachers in their profession and improve both the teaching and learning in their classroom. The next section will detail implementation of our decade-long program along with data collected and analyzed by independent evaluators.

3. IMPLEMENTATION & KEY FINDINGS

While the results of our CPACK professional development program have already been documented in earlier publications, such as Yaşar *et al.* (2014), their importance for and relevance to the aforementioned theoretical frameworks have gradually come to our attention in recent years as a result of our work in pedagogy and cognitive sciences. In this section, we briefly review findings on teaching and learning that are relevant to our discussion.

While the main activity of our study has been teachers' computational pedagogical content knowledge professional development, the ultimate desired outcome was better student engagement and learning as well as teacher engagement/retention and teaching. A mixed-methods approach (Creswell 2012) was used to collect and analyze qualitative data (interviews, activity logs, observations, pre- and post-activity surveys, and artifacts) as well as quantitative data (student grades and report cards, test scores, and standardized exams by the NY State) for the purpose of formative and summative assessment.

Integration of modeling and simulation tools, such as those in Table 1, into secondary school teaching was initially done in three steps by incrementally adding a new domain of knowledge each year for the first three years. As shown in Table 2, the first step of the multi-tier incentive-based PD included technological knowledge (TK) training, the second step included technological content knowledge (TCK) training, and the final step included teaching of content through computational and pedagogical tools. Here, technology knowledge (TK) means knowledge of technology tools and their use. Technological content knowledge (TCK) means integrating knowledge of technology and STEM (physics, chemistry, biology, math, etc.) for the purpose of teaching its content. Technological pedagogical content knowledge (TPCK) means applying pedagogical technologies to the teaching of STEM content.

Table 2. Profiles of teachers from Urban (U) and Suburban (SU) School Districts at the CPACK summer training (2003-2007).

Training→	TK		TCK		TPCK		Total
School →	U	SU	U	SU	U	SU	
Math	96	14	42	2	22	0	176
Science	38	15	17	9	12	5	96
Tech	7	3	5	1	2	1	19
Special Ed	14	1	2	0	1	0	18
TOTAL	155	33	66	12	37	6	309

Supported by the National Science Foundation through various grants, we formed a CMST Institute in 2002 and have since been offering CPACK PD to in-service and pre-service secondary school teachers. The professional development program has both summer and academic-year components. While we constantly explore new tools, we continue to use those in Table 1 because of a large database of artifacts and lesson plans we have developed using them over the past decade. Table 2 shows the number of in-service teachers who benefited from the summer institute component offered through NSF support in partnership with local school districts (Rochester City School District (RCSD) and Brighton Central School District (BCSD)) and several national organizations (Shodor Foundation, Krell Institute, and Texas Instruments). Almost half of teachers who attended TK training returned for additional TCK training, and half of those returned for TPCK training. This is typical of an incentive-based PD (Loucks-Hersley *et al.* 2010). Teachers have multiple summer engagements and some teach in district summer schools. So, the dates and time impact attendance. For those who could not attend due to such circumstances, we offered similar short courses during the school year. The partnering districts also offered a condensed version of the training to additional 160 teachers through turnkey training and PD days. For the purpose of gathering data for research and evaluation, we only worked with teachers who attended the summer institute as part of commitment to the study.

The initiative displayed elements of a scalable innovation (Dede *et al.* 2005), especially in mathematics. There was a cultural change in all 15 secondary schools at the urban RCSD and the suburban BCSD. They were fully engaged all the way from superintendents and principals down to teachers and students. Improved teacher retention and student achievement reported by partnering districts drew national attention to this initiative, including testimony by the author, Jeff Mikols (a RCSD math teacher who is now a district curriculum director), and Ed Chi (a BCSD science teacher who has left the district) before the U.S. Congress (House Hearing 2003).

In a 2010 survey of 40 TCK and TPCK teachers, 94% agreed that the training made them more effective in the classroom; 87% agreed that it strengthened their pedagogical skills; 73% agreed that it strengthened their pedagogical content knowledge; 100% agreed that training strengthened their skills related to modeling and simulation; 86% reported that they continue to use the hardware, software and other materials made available through the project in their classrooms; and 80% believed that their participation served to build leadership skills. Seven years after the start of the initiative, 73% of participating teachers at RCSD were still teaching while 10% had moved to lead positions (Yaşar *et al.* 2014). According to the National Center for Education Statistics (NCES 2014), about 16% of STEM teachers either move to another school or leave the profession every year. The national average is that nearly half of all new STEM teachers leave the job within five years (Graziano 2005). Although we do not have the 2002 baseline data from participating districts to compare with, urban schools such as RCSD generally perform much worse than the national average. RCSD district officials reported throughout the initiative (Crowley 2007) that it not only helped retain veteran teachers but it also drew more and better teachers to an urban school district, which usually has a hard time recruiting teachers because of the well-known urban problems (Margolis *et al.* 2008).

Table 3: Frequency of technology tools used by trained teachers.

Subject /Grade	Daily	Weekly	Bi-weekly	Special Projects
Math Grades 7-8	Laptop, smartboard	Power Point, PI, TI tools, GSP, Excel, Flash	AgentSheets	Interactive Physics (IP), Stella, Java, GIS/GPS
Math Grades 9-12	Laptop, smartboard, TI tools	Power Point, PI	Excel, Flash	IP, Stella, Java, GIS/GPS
Science Grades 7-8	Laptop, smartboard, Power Point	AgentSheets, Excel, PI	TI, GIS/GPS, Flash, Java	Stella, GSP, Interactive Physics (IP)
Science Grades 9-12	Laptop, smartboard	Flash, Excel, Power Point	Interactive Physics (IP), Java, GPS	Stella, AgentSheet, GIS, PhET

All of the trained secondary school (grades 7-12) teachers reported that on a daily base they used laptops for presentations, graphing calculators for math instruction, and electronic smart boards for interactive lessons (see Table 3). Positive experience with C-MST tools is believed to have initiated use of additional tools such as GIS/GPS, Java, Flash, and PhET (Wieman *et al.* 2008). Annual surveys of teachers showed that usage of the tools in the classroom was directly linked to the amount of training they had received. In post-training journals, while only 60% of the teachers reported occasional use of modeling tools in their classrooms after the initial TK training, 78% reported that they used them <u>regularly</u> after the TPCK training.

Table 4: Percent of teachers using modeling in class

Grade Level	Frequency		
	Regularly	Special Projects	No
7-8 Math	46%	46%	8%
9-12 Math	60%	35%	5%
7-8 Science	25%	75%	25%
9-12 Science	54%	38%	8%

In a 2007 survey by 65 active teachers who had received at least two years of training, many reported a significant use of modeling tools for both classroom instruction and special projects (see Table 4). It appears that the higher the grade level, the more regularly these tools are used in the classroom. Less frequent use of tools in RCSD middle school science classes was a concern, which resulted from access and scheduling problems but it got better over time as the concern was conveyed to the district administration. At BCSD, access to computing resources was not an issue. For example, participating teachers ended up fully integrating Interactive Physics into their high school physics labs.

Figures 5 through 8 show some of the survey results in graphical format regarding student engagement and learning as a result of CMST-enhanced teaching. More than 92% of surveyed teachers agreed that computational inquiry made math and science concepts significantly more comprehensible to students (Fig. 5).

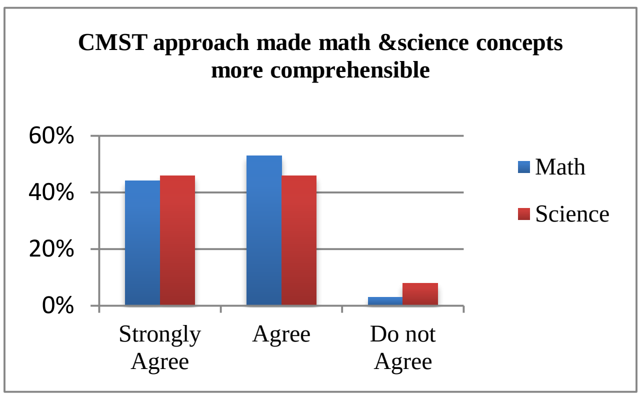

Figure: 5. Improved comprehension of STEM concepts.

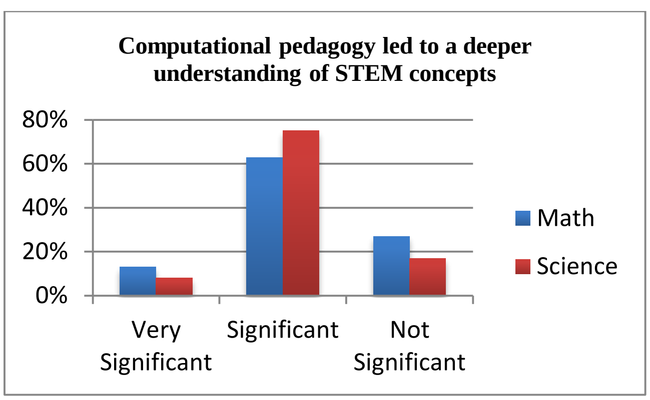

Figure 6: Deeper understanding of STEM concepts.

100% of technology, 72% of math, and 31% of science teachers reported observed improvement in students' problem solving skills. Student reaction to modeling (versus traditional techniques) was found to be 97% favorable in math and 77% in science classes. While science classes utilized technology less due to limited access and lack of science-related modeling examples, in instances where it was utilized, a deeper understanding of science topics was achieved, compared to math topics (83% vs. 76%, see Fig. 6). As seen in Fig. 7, students in higher-grade levels found computational modeling more engaging in both math classes (grades 7-8: 77% vs. grades 9-12: 90%) and science classes (grades 7-8: 75% vs. grades 9-12: 85%). Modeling was even found helpful to non-traditional (special education) learners (Fig. 8); again the higher the grade level the higher the engagement: math classes (grades 7-8: %76 vs. grades 9-12: 100%) and science classes (grades 7-8: 75% vs. grades 9-12: 85%).

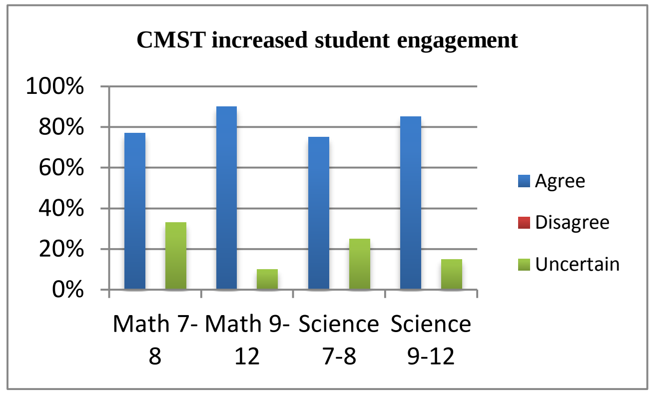

Figure 7: Student engagement per grade level and subject.

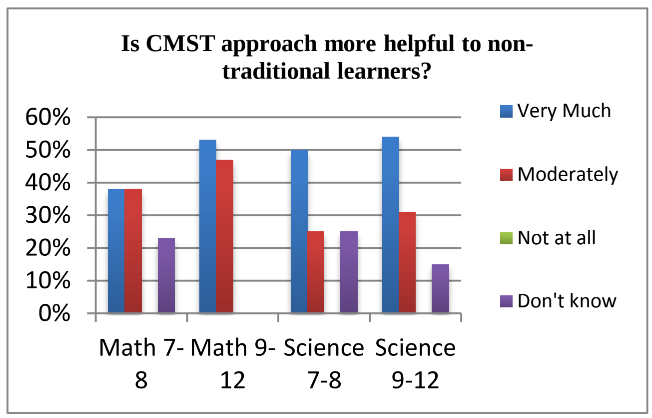

Figure 8: Impact on non-traditional learners.

Qualitative data from journal entries, activity logs, and teacher interviews pointed out to an emerging pattern regarding gender response to CMST-based teaching. Two independent coders read the 2010 teacher survey data and coded the text segments to arrive at descriptions and common themes. An inductive process (Creswell 2012) was used to group these codes in order to form even broader themes. Based on detailed accounts of 26 teachers (out of 40), the evaluators arrived at the following broad theme:

> *While male students showed more interest in playing with technology and plowing through the details with less regard to the big picture, female students initially seemed reluctant and timid but excelled when details (curriculum) were put into context of real-world problems and projects.*

This is consistent with national findings by our collaborators such as Repenning (2012). It is also consistent with our own data when triangulated against student scores and graduation rates. For example, while cohorts of 8th grader male and female students from both districts had a gap in their average math performance at the beginning of the initiative, not only were the gaps closed but also reversed four years later (12th grade) as shown in Table 5. At RCSD, while both male and female students did much better than four years earlier, the graduation rate of the same cohorts still reflected a gender-based trend in performance growth, favoring female students. To examine whether the difference is statistically significant, we calculated the z-scores assuming a normal distribution approximation (Brase & Brase 2012). The sample sizes for male and female students were roughly the same at both districts, with about 1200 at RCSD and 150 at BCSD. The column p indicates the confidence level that the difference between males and females may be due to a nonrandom effect. Normally, any confidence level below 90% is less than significant. Here, with more than 90% confidence level female cohorts outperformed male cohorts in both math performance and graduation rates.

Table 5: Gender-based performance history at RCSD & BCSD.

		2001-2002		2005-2006			
		Gender		Gender		Z	P
		M	F	M	F	score	(%)
RCSD	Math Cohort	13%	10%	41%	49%	3.97	99
	Graduation Rate			34%	44%	5.06	99
BCSD	Math Cohort	92%	84%	93%	93%	0	0
	Graduation Rate			85%	90%	1.29	90

To further triangulate self-reporting data by teachers, annual student achievement data were analyzed in the partnering school districts via report cards and standardized test scores. While we cannot fully isolate the impact of teacher training from other contributing factors, an upward district-wide trend was noted in both urban and suburban districts during the initiative. The percentage of students receiving a Regents diploma increased significantly from the baseline (RCSD: 21% → 59%, BCSD: 84% → 95%). The initiative exposed students from the urban district to college experiences and opportunities, and this may have led to an increased interest (78% → 83%) in both 2-year and 4-year college enrollments over the period examined. Furthermore, the passing rate (>65/100) in NY State Grade-8 Math exam increased in Rochester City SD from 10% to 33%, while the passing rate in NY Regents Math-A exam (Grade 11-12) also increased from 13% to 67%. Passing rate in sciences also increased in areas such as Physics (3% → 22%) and Chemistry (9% → 27%). At BCSD, passing rates improved in mathematics (Math-A: 51% → 99%) and sciences (Physics: 52% → 78%). The number of students taking General Physics at Brighton increased from 50% to ~100% and the number of students taking AP Physics also doubled. Student passing rates at both districts seemed to reflect relative participation of district's math and science teachers in the initiative. All of the improvements have been found to be statically significant for typical sample sizes from each district.

The main goal of the sponsoring *No Child Left Behind* program was to train as many teachers as possible to potentially create a district wide impact on student achievement scores. As a result we trained twice as many as we had committed to (see Table 2). While the goals of the sponsoring agency were met, as witnessed by gains in the standardized test scores reported by partnering districts, no comprehensive research was done by the project to more closely link the gains in student achievement scores to the teaching and learning resulted from the initiative. By the time the goals of sponsoring NSF program shifted from 'leaving no child behind' outreach to 'researching the interventions' we had almost run out of control groups in partnering school districts' math classrooms. The initiative invited science teachers but limited access to computer labs, skepticism about use of technology, and inadequate number of readymade curricular modules discouraged many to invest in trainings that lacked significant science content and representative lesson plans. By the end of the project while almost all secondary math teachers in RCSD and BCSD received training and yearlong PD, only 20% of science teachers took part.

In final years of the study, when focus shifted towards researching the intervention, a few treatment-control comparisons were conducted. A pair of CMST and non-CMST high school teachers from the same school taught properties of quadrilaterals in a mathematics class. The CMST teacher used GSP in a class of 24 pupils while the non-CMST teacher used conventional methods in a class of 14 pupils. Both teachers conducted the same unit test. Even though the CMST teacher taught a more crowded class, his classroom average was 82.5 versus 49.5 for the other class. The second study involved a math triathlon similar to Regents Math A and B tests involving use of TI graphing calculators. Scored by external judges, including teachers and college faculty, this study revealed that students taught by CMST teachers outperformed other students in all categories: Math-A: 60.26 vs. 49.54; Math-B: 71.9 vs. 55.6; and 7-8 Grade Math: 64.0 vs. 58.6.

Over the past decade, institute staff and participants created a large database of more than 300 CMST curriculum modules and lesson plans. Curriculum modules and lesson plans from the database have been downloaded by people around the world at a rate of 50-80 per day, totaling almost 100,000 since the database was launched. The database has also provided content for two local pre-service methods courses (NAS 401/501 C-MST Tools and NAS 402/502 Computational Pedagogy) in the college's teacher education program. Table 6 shows pre-service enrollments in these credit-bearing NAS courses. Additionally, the database supported turnkey training offered by partnering districts during professional development days, serving 160 in-service teachers.

The CMST database (www.brockport.edu/cmst) continues to support three general education courses reported earlier in this journal (Yaşar 2013). They have since served 500 more STEM undergraduates. The two NAS methods and 3 general education courses have become part of the NSF Robert Noyce Scholarship program since 2012, serving a new cadre of 50 computationally competent STEM teachers, some of whom have already started teaching in high needs school districts both locally and nationally.

Table 6. Number of pre-service teachers trained.

Courses	2003-07	2008-12	2013-15	Total
C-MST Tools & Pedagogy	113	107	105	325

In Rochester City and Brighton Central secondary school classrooms taught by CMST teachers, students were all given a chance to experience the deductive and inductive learning processes. As mentioned earlier, 97% of mathematics and 92% of sciences classes using the CMST approach agreed that it made subject-related concepts more comprehensible. Furthermore, 83%

of science classes and 76% of math classes found that it led to even a deeper understanding of STEM concepts. While modeling is a common practice in mathematics and science classes, science classes often go beyond modeling to utilize simulations in order to investigate time-dependent dynamics of scientific phenomena. When used together, modeling and simulation affords the learner a constructivist opportunity (Grabinger & Dunlap 1995) to cycle iteratively back and forth between the inductive and deductive approaches to learning (Yaşar & Maliekal 2014). Teaching mathematical and computing concepts contextually has been recommended for quite some time by national learning standards (NGSS, Computing Curriculum 2005) but we now additionally know from cognitive sciences that retrieval practices attempted at various contexts is good — because every time the recall is attempted in a different context, it establishes more links that will help the remembering and learning (Brown *et al.* 2014).

Benefits of constructivist and contextual learning was observed in an annual after-school CMST challenge competition, which allowed students more time and freedom than a regular classroom setting to apply, test, and revise the constructed computational models. Participating students had a full semester to develop a team project. Scoring rubric included problem statement, application of the model to a problem of interest, data analysis, teamwork, originality, electronic demonstration, and presentation of the results before a panel. Extra points were given for use of multiple CMST tools, demonstrated understanding of computational, mathematical and scientific content, and level of challenge, knowledge and skills demonstrated beyond team's grade level. As expected, the incentives helped push students to go beyond initial job of model construction, playful experimentation, and introductory exposure to STEM concepts. A project-based experience reported in Yaşar *et al.* (2005) by a group of 9th grade high school students from Brighton High School (NY), who used the Interactive Physics and Geometer's Sketch Pad to prove Kepler's Laws in an afterschool program (annual CMST Challenge), is a testimony of how students gained a deeper understanding of computational and scientific content of the planetary motion. Following is a sentiment by these high school students after their CMST experience to prove Kepler's laws:

> *"We had not taken any physics courses and we were not fully knowledgeable about laws of universe that govern planetary motion. That is not different from the situation of Kepler; as no one quite knew how gravitational forces worked until Newton came. Kepler had access to data compiled by Tycho Brahe and he looked for patterns. We had access to modern tools and we looked for miracles! We learned how to transfer visuals images and data from Interactive Physics to Geometer's Sketchpad to measure angles, distance, and areas of triangles needed for the proofs… While it was initially frustrating to learn new tools, realizing what Kepler would have done if he had such tools; we quickly learned to appreciate the opportunity in our hands. In the end, we did not make a discovery in physics, but we certainly discovered that physics was not a threatening or boring subject. We also discovered the role of mathematics in physics. The foreboding nature of complicated physics was abolished and we all looked forward to taking physics classes."*

The authors followed progression of these students as a case study. In their project the following year, these 10th graders inquired further about fundamental STEM principles of their projects and operational principles of the tools they used for modeling and simulations. Using Excel to compute a simple algebraic form of *rate of change* equation, *new = old + change*, that they had learned in the mathematics class that year, they attempted to replicate the Interactive Physics results found earlier for the harmonic and planetary motion. For the harmonic motion in Fig. 4, this involved computing algebraic formulas for the position ($x_{new} = x_{old} + dx$) and velocity ($v_{new} = v_{old} + dv$) of the spring-driven object at times ($t_{new} = t_{old} + dt$) separated by interval dt. While time (t) was an independent variable, and *change* in x was dependent on the velocity as $dx = v \cdot dt$, and the change in v was dependent on the acceleration as $dv = a \cdot dt$, where acceleration (a) is *Force/mass.* The force applied by a spring unto an attached box is $F = - k \cdot x$, where k is the stiffness coefficient of the spring and x is the displacement of the box from the equilibrium position (x=0). The details of their self-constructed simulations is given in Yaşar *et al.* (2006), yet the brief statement below summarizes the progress they had made — they were no longer threatened or frustrated by learning of science.

> "*Through Excel, we were able to use a simple algebraic equation (new = old + change) to manually construct our own simulations as an alternative way and compared them to those done earlier by the Interactive Physics. To compute the "change" all we needed was some basic knowledge of the force that governed the system, whether it was the harmonic or the planetary motion."*

The progression by these students show that the learner can start either with a readymade model, or construct one using a pull-down menu, that represents the scientific phenomenon under study and conduct fun experiments without having to know the details of the model and the laws that govern its motion. If it stops there, then we can say that the top-down deductive approach has engaged students in STEM activities. But, if the learner is tempted to continue and inquire about the initial model's constitutive parts and forces that act on them, then he can run simulations by changing characteristics of the parts and forces to inductively construct a new model and physical setting that better represent the reality. This cycle can be repeated until the desired knowledge or outcome is reached. This way of learning, through inquiry and experience, is nothing but how scientists do their work (Bransford *et al*. 2000, Donovan & Bransford 2005). Such an iterative and stepwise progression in constructive learning is also consistent with several pedagogical frameworks, including scaffolding, zone of proximal development (ZPD) that we discussed earlier by Vygotsky, and the *Optimal Flow* (Csikszentmihalyi 1990) shown in Fig. 9, which suggests the importance of balancing challenges and abilities using pedagogical stepping-stones in order to attain optimal flow for a learner.

Figure 9: Illustration of Optimal Flow as a path of learning.

4. CONCLUSION

Cognitive psychology research has shown that interleaved retrieval practice has great advantages for gaining deep and lasting knowledge. Interdisciplinary education is a form of this practice at course and curriculum levels but it takes effort and time, thereby slowing down the learning process. In college, it delays graduation and in K-12 it slows down the pace of teaching. Technology can speed it up but this throws another ranch into the works by adding another domain of knowledge. So, the question becomes of finding a technology that will facilitate mixing of multiple views around a topic in a pedagogically way. This, we claim, calls for the use of computational modeling and simulation technology because it naturally adds a deductive and inductive pedagogy to teaching of STEM content. The final question, then, becomes, "OK, we got this wonderful thing, how do we go about institutionalizing it?' And, this is where the need for teacher training becomes the central task, as they are the agents of change for any reform in the schools (Bybee & Loucks-Hersley 2000, Loucks-Hersley *et al.* 2010).

We have run a decade-long experiment to study the task explained above, using CMST tools within an interdisciplinary CPACK framework for teacher professional development. Triangulated data from multiple sources indicated that the use of CMST tools and pedagogy not only supported basic interleaved retrieval practices but it enriched such practices by putting the learner on the driver seat through an iterative cycle of constructivism, interactivity and immediate assessment. Not only did this cyclic process helped students: a) engage in a topic through a general simplistic introduction and b) move deeper deductively into more content as they gained more skills, but it also enabled them to construct significant knowledge through easy experimentation to inductively draw conclusions about the topic they started with. Computational modeling and simulation involves all of these as demonstrated in our initiative in public schools. The deductive aspect of modeling helped teachers present science concepts to learners by simplification of reality, which was instrumental to draw young minds into science learning. High levels of student engagement reported by our participating teachers strongly support the effectiveness of computational modeling as a deductive pedagogical tool. The CMST tools did exactly as expected by shielding students from having to know detailed content knowledge of mathematics (e.g., differential equations), computing (e.g., algorithmic and programming) and science (e.g., physics) to conduct experiments of linear, harmonic, and planetary motion using IP. The inductive process resulting from experimentation through simulations helped learners to rediscover principles of computing and sciences, therefore leading to deeper content learning. Since it is the inductive reasoning that help us come up with general patterns and simplifications from paralyzing details, one cannot have a chance to utilize a deductive approach if there had not been an inductive counteract to simplify concepts for later use. So, we do not have an option of choosing one over the other in education; we need to use both, as they complete — not compete with — each other. Improved student achievement scores in both local and statewide exams at partnering school districts point out to a lasting impact of the dual nature of computational pedagogy.

Our initial focus on pedagogical aspects of CMST was to develop a tool-independent CPACK training for our teacher education program in order to maximize transfer of curriculum inventories to new conditions when newer technologies become available. However, we stumbled upon much more. Information revolution has taken electronic computing devices to every corner of the globe but very few would be familiar with and relate to computational modeling and simulation. In fact, even some researchers and educators might consider CMST as an *ad hoc* technology. Computing is not usually considered as a branch of science (Denning 2009) because it deals with artificial phenomena, not natural phenomena. However**,** as artificial and imitational as electronic computation is, it might actually help us discover how the biological computation generates complex mental states. We think it is going to do more than that, as understanding how pervasive the computational behavior is might change the way we relate to ourselves and everything else in the universe.

Computational theory of mind considers electronic and biological computing devices to compute the same way at the fundamental level, but much is needed to reduce our complex mental states to mere computational processing of information. Regardless of what high level processes a computing device is performing, we think that the way computing is done at the most fundamental level will carry itself all the way to the top level. Computational modeling and simulation is a high level electronic process whose dual characteristic does reflect the two fundamental modes of computing (i.e., addition and subtraction). Deductive and inductive thinking, on the other hand, are also two high-level cognitive processes that similarly reflect the same modes of computation. So, one can suggest that it is the computable nature of information that leads to commonality of electronic and cognitive outcomes of computing regardless of the underlying structure. A million-dollar question would then be 'what is the source of information's computable (associative and distributive) behavior?' Is it merely reflecting how the matter itself behaves?

Computability actually appears to be a universal characteristic of both granular matter and quantifiable information. Anything quantifiable has three distinguishable outcomes: quantity, sequence, and pattern. If quantifiable stuff ─ be it matter or information ─ can form various patterns to make up atomic and cellular structures as well as instructions and thoughts, then everything we see out there is computable (Montague 2006; pp. 14). If so, then perhaps we can start examining a computational theory of everything (Yaşar 2016) that would mean everything in the universe behaves computationally by either uniting with (addition) or departing from (subtraction) other things to form a new sum as, again, depicted in Fig. 1. Our current and future studies will continue along these lines. Any traction that it might gain will be a tribute to Turing.

5. ACKNOWLEDGEMENT

This work was supported by the National Science Foundation (NSF) funds via Grants EHR-0226962, SCI-0520036, DUE-0942569, and DUE-1136332. We would like to thank to faculty and teachers whose efforts contributed to the development, teaching, and assessment of the reported courses and materials.

6. REFERENCES

[1] Armoni, M. (2013) "On Teaching Abstraction to Computer Science Novices." *J. Comp in Math & Science Teaching*, 32 (3); 265-284.

[2] Bell, L. R., Gess-Newsome, J. and Luft, J. (2008) Technology in the Secondary Science Classroom. *National Science Teachers Association (NSTA).*

[3] Bell, L. R and Smetana, L. K. (2008). Using Computer Simulations to Enhance Science Teaching and Learning. In Technology in the Secondary Science Classroom (Eds. Bell et al.). Washington, DC: NSTA Press.

[4] Bransford, J., Brown, A. and Cocking, R. (2000). *How People Learn: Brain, Mind, Experience, and School*. National Academy Press, Washington, D.C.

[5] Brase, C. H. and Brase, C. P. (2012). *Understandable Statistics.* 10^{th} Edition. ISBN: 0840048386.

[6] Brown, P. C., Roediger, H. L. and McDaniel, M. A. (2014) *Make it Stick.* The Belknap Press of Harvard University.

[7] Bybee, R. W., and Loucks-Horsley, S. (2000). Advancing Technology Education: The Role of Professional Development. *The Technology Teacher*, Oct. 2000, 31-34.

[8] Creswell, J. W. (2012) *Educational Research: Planning, Conducting and Evaluating Quantitative and Qualitative Research*. 4th Edition. Pearson Education, Inc.

[9] Computing Curricula. 2005. A Cooperative Project of the Association for Computing Machinery, the Association for Information Sciences, and the IEEE Computer Society. http://www.acm.org/education.

[10] Crowley, M. (2007). Rochester City School District. Mathematics Curriculum Director. Private communication. Email: Margaret.Crowley@rcsdk12.org.

[11] Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. New York: Harper Collins.

[12] Dede, C., Honan, J., and Peters, L. (2005). *Scaling Up Success: Lessons Learned from Technology-Based Educational Improvement.* John Wiley and Sons.

[13] De Jong, T., & Van Joolingen, W. R. (1998). "Scientific Discovery Learning with Computer Simulations of Conceptual Domains." *Review of Educational Research*, 68(2), 179-201.

[14] Denning, P. (2009). "Beyond Computational Thinking." *Communications of the ACM*, Vol. 52 No. 6, Pages 28-30.

[15] Donovan, S. and Bransford, J. D. (2005). *How Students Learn.* The National Academies Press, Washington, D.C.

[16] Grabinger, R. S. and Dunlap, J. C. (1995). "Rich environments for active learning: a definition," *Association for Learning Technology*, **3** (2); 5-34.

[17] Graziano, C. (2005). "Public Education Faces a Crisis in Teacher Retention." *Edutopia*, Feb. 9, 2005. http://www.edutopia.org/new-teacher-burnout-retention.

[18] Geometer's Sketch Pad (GSP). http://www.dynamicgeometry.com.

[19] Flick, L. and Bell, R. L. (2000). "Preparing tomorrow's science teachers to use technology: guidelines for Science educators." *Contemp Issues Technol Teach Educ* 1: 39-60.

[20] Hammond, L-D., Austin, K., Orcutt, S. and Rosso, J. (2001). "How People Learn: Introduction to Learning Theories. Stanford University." http://web.stanford.edu/class/ed269/hplintrochapter.pdf.

[21] House Hearing (2003). *Implementation of The Math and Science Partnership Program: Views From The Field.* 108th Congress. https://www.gpo.gov/fdsys/pkg/CHRG-108hhrg90162/pdf/CHRG-108hhrg90162.pdf. Oct. 30, 2003.

[22] Illeris, K. (2009). *Contemporary Theories of Learning*. Taylor & Francis Group: New York.

[23] Koehler, M. J., and Mishra, P. (2008). "Introducing TPCK," in *Handbook of Technological Pedagogical Content Knowledge (TPCK) for Educators*, Routledge Press, New York & London.

[24] Koffka, K. (1935). *Principles of Gestalt psychology*. New York: Harcourt, Brace & World.

[25] Landau, R. (2006). "Computational Physics: A Better Model for Physics Education?" *IEEE Comp. in Sci & Eng.*, 8 (5), 22-30.

[26] Little, L. (2003). "The computational science major at SUNY Brockport." *FGCS*, 19, pp. 1285-1292.

[27] Loucks-Horsley, S., Stiles, K. E., Mundry, S., Love, N., Hewson. (2010). *Designing professional development for teachers of science and mathematics*. Third Edition, Thousand Oaks, CA: Corwin Press.

[28] MacDonald, M. (2008) *Your Brain: The Missing Manual*. O'Reilly Media: Canada.

[29] Margolis, J., Estrella, R., Goode, J., Holme, J. and Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing*. Cambridge, MA: MIT Press.

[30] Mishra, P., Koehler, M. J. (2006). "Technological pedagogical content knowledge: A framework for integrating technology in teacher knowledge." *Teachers College Record*, 108 (6), 1017-1054.

[31] Montague, R. (2006). *Your Brain Is (Almost) Perfect: How We Make Decisions.* Plume Books: New York.

[32] Mooney, C. G. (2013). *An Introduction to Dewey, Montessori, Erikson, Piaget, and Vygotsky*. Redleaf Press: St. Paul, MN, 2013.

[33] NCES (2014). "Teacher Attrition and Mobility: Results From the 2012-13 Teacher Follow-up Survey." National Center for Education Statistics, Sept. 2014. Report No. NCES 2014077.

[34] NRC (2012). *A framework for K-12 science education: practices, crosscutting concepts, and core ideas*. National Research Council Report. National Academies Press, Washington.

[35] NGSS. Next Generation Science Standards. http://www.nextgenscience.org.

[36] Niess, M. (2005). "Preparing teachers to teach science and mathematics with technology: Developing a technology pedagogical content knowledge." *Teaching and Teacher Education*, 21, 509-523.

[37] NSF Blue Ribbon Report (2006). *Simulation-Based Engineering Science: Revolutionizing Engineering through Simulation.* http://www.nsf.gov/pubs/reports/sbes_final_report.pdf.

[38] Osborne, J. and Dillon, J. (2008). *Science Education in Europe: Critical Reflections*. A Report to The Nuffield Foundation. Retrieved November 12, 20015. http://www.nuffieldfoundation.org/sites/default/files/Sci_Ed_in_Europe_Report_Final.pdf

[39] PITAC Report (2005). "Computational Science: Ensuring America's Competitiveness," http://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf.

[40] Project Interactivate. http://www.shodor.org/interactivate/. Supported by NSF CISE funds.

[41] Prince, M. J. and Felder, R. M. 2006. “Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases.” *J. Engr. Education,* **95** (2); 123-138.

[42] Prince, M. J. and Felder, R. M. 2007. "The Many Faces of Inductive Teaching and Learning." *Journal of College Science Teaching*, **36** (5); 14-20.

[43] Reed, D. & Dongarra, J. (2015). “Exascale Computing and Big Data.” *Communications of the ACM*, 58 (7), pp. 56-68.

[44] Repenning, A. 2012. “Programming Goes Back to School.” *Communications of the ACM,* 55 (5), 35-37.

[45] Restak, R. (2001) *The Secret Life of the Brain.* The Dana Press: New York.

[46] Rutten, N., van Joolingen, R., and van der Veen. (2012). “The Learning Effects of Computer Simulations in Science Education.” *Computer & Education*, 58; 136-153.

[47] SIAM Report on Graduate CSE Education (2001), *SIAM Review*, 43, pp. 163-177.

[48] SIAM Report on Undergraduate CSE Education (2007), SIAM Conference on CSE, www.siam.org/conferences, February 19-23, 2007, CA.

[49] Sjøberg, S. and Schreiner, C. 2005. “How do learners in different cultures relate to science and technology? Results and perspectives from the project ROSE.” (http://roseproject.no). Asia Pacific Forum on Science Learning & Teaching, 6, 1-16.

[50] Smetana, L. K. and Bell, R. L. (2012). “Computer Simulations to Support Science Instruction and Learning: A critical review of the literature.” *Int. J. Science Education*, 34 (9); 1337-1370.

[51] STELLA. http://www.iseesystems.com.

[52] Swanson, Charles. (2002). “A Survey of Computational Science Education.” *http://cssvc.ecsu.edu/krell/Computational%20Science%20Education%20Survey%20Paper.htm*.

[53] Tenenbaum, J. B., Kemp, C., Griffiths, T. L. & Goodman, N. D. (2011). “How to Grow a Mind: Statistics, Structure, and Abstraction.” *Science,* 331, 1279-1285.

[54] Turing, A.M. (1936). "On Computable Numbers, with an Application to the Entscheidungs problem." *Proceedings of the London Mathematical Society.* 2 (1937) 42: 230–265. doi:10.1112/plms/s2-42.1.230.

[55] Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

[56] Wieman, C. E., Adams, W. K. and Perkins, K. K. (2008). “PhET: Simulations That Enhance Learning.” *Science*, 332, 682-83.

[57] Wing, J. M. 2006. “Computational Thinking,” *Communications of the ACM*, Vol. 49, No. 3, 33-35.

[58] Yaşar, O. (2016). “Computational Theory of Everything.” To be submitted. Under preparation.

[59] Yaşar, O. (2015). “A Universal Process: How Mind and Matter Seem to Work.” *Science Discovery,* 3 (6), 79-87.

[60] Yaşar, O., Veronesi, P., Maliekal, J. and Little, L. (2015) “Computational Pedagogical Content Knowledge (CPACK).” In D. Slykhuis & G. Marks (Eds.), *Proceedings of Society for Information Technology & Teacher Education Conference 2015* (pp. 3514-3521).

[61] Yaşar, O. (2014). “A Pedagogical Approach to Teaching Computing Principles in the Context of Modeling and Simulations.” *J. Computing Teachers,* Winter Issue.

[62] Yaşar, O. and Maliekal, J. (2014a). “Computational Pedagogy: A Modeling and Simulation Approach.” *IEEE Comp. in Sci & Eng*, **16** (3), 78-88.

[63] Yaşar, O. and Maliekal, J. (2014b). “Computational Pedagogy Approach to STEM Teaching and Learning.” In M. Searson & M. Ochoa (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2014* (pp. 131-139). Chesapeake, VA: AACE.

[64] Yaşar, O., J. Maliekal, L. little, and P. Veronesi. (2104). “An Interdisciplinary Approach to Professional Development for Math, Science, and Technology Teachers.” *Journal of Comp. in Mathematics and Science Teaching*, 33 (3), pp. 349-374.

[65] Yaşar, O. (2013a). “Computational Math, Science, and Technology (C-MST) Approach to General Education Courses.” *J. Computational Science Education,* 4 (1), 2-10.

[66] Yaşar, O. (2013b). “Teaching Science through Computation.” *Science, Tech and Society,* Vol. 1 (1), 9-18.

[67] Yaşar, O. and R. Landau. (2003). “Elements of Computational Science and Engineering Education.” *SIAM Review,* 45, pp. 787-805.

[68] Yaşar, O. (2001). “Computational Science Education: Standards, Learning Outcomes and Assessment.” *Lecture Notes in Computer Science*, 2073, 1159-1169.

[69] Yaşar, O., Rajasethupathy, K., Tuzun, R., McCoy, A. and Harkin, J. (2000). “A New Perspective on Computational Science Education,” *IEEE Comp. in Sci & Eng*, **5** (2), 74-79.

[70] Yaşar, P., Kashyap, S., and Roxanne, R. (2005). “Mathematical and Computational Tools to Observe Kepler’s Laws of Motion.” NSF MSPNET Library, http://hub.mspnet.org/index.cfm/14566.

[71] Yaşar, P., Kashyap, S., and Taylor, C. (2006). “Limitations of the Accuracy of Numerical Integration and Simulation Technology.” NSF MSPNET Library. http://hub.mspnet.org/index.cfm/14568.