Constructionism and Deconstructionism

Pavel Boytchev • Sofia University, Bulgaria • boytchev/at/fmi.uni-sofia.bg

> Context • There is a movement to change education so that it is adequate to social expectations and uses the full potential of technology. However, there has been no significant breakthrough in this area and there is no clear evidence why. **> Problem** • A potential issue explaining why education falls behind is the way educators focus on education. There is a possibility that a significant step in the learning process is routinely neglected. **> Method** • Two different approaches to using IT in education are tested in two different environments: a university level course based on constructionism and IBL projects for secondary school students. **> Results** • It is possible to apply constructionism in education, but there are still problems. They are not related to how students construct knowledge, but how they deconstruct knowledge. **> Implications** • The most significant problem of deconstruction is that it requires creative skills. This makes it very difficult to formalize it and to provide effective recommendations for its application. **> Constructivist Content** • Deconstruction is a prerequisite of construction, thus deconstructionism deserves more attention and study. A proper application of deconstructionism will make it possible to reconstruct education in a way that is impossible with the current approaches. **> Key words** • **Deconstructionism, constructionism, future of education, inquiry-based learning.**

Constructivism and constructionism

« 1 » Constructivism in education is a philosophy that advocates the construction of knowledge through real-life or real-life-like experiments fostering learning. The role of the teacher is not to transmit or to impose knowledge, but to guide the learner through his personal journey in learning.

« 2 » The earliest examples of constructivism in education were proposed by John Dewey and Maria Montessori. Dewey (1910) described thinking as a natural act that should be supported by an encouraging environment that is rather different from the monotonous uniformity of classrooms and textbooks. An important factor for the development of creative thinking is the curiosity that leads to exploration. According to Montessori, education starts from birth and "[t]he child must not be considered as he is today […] He must be considered in his power of potential man" (Montessori 2001: 3). She built unique learning environments that are considerate of the student's physiological and psychological age.

« 3 » A significant contribution to constructivism was made by Jean Piaget. He saw learning as a continuous process where a student assimilates knowledge entities into meaningful knowledge constructs. Constructivism, as described by Piaget, is focused on the mental models of the world. This theory was further extended by Seymour Papert in a way that applies it to practical construction. Papert called this "constructionism." The main concept is that constructing tangible artefacts helps the construction of mental understanding of the world. Papert proposed an extensive use of IT in the classroom that supports another important aspect of constructionism, namely, that constructing entities is public in the sense that they are observable by others. More importantly, the process of construction is also public and this makes learning more effective and sustainable.

Constructionism in education

Constructionism at university level

« 4 » The concepts of constructionism have been applied to education and the results are promising. Several courses introduced by the Faculty of Mathematics and Informatics at Sofia University are focused on educational software and real-time computer animation (Boytchev 2007). In these courses, students learn the basic skills and approaches of building complex constructs out of a small set of elements. One of these courses is Geometry of Motion – a multi-disciplinary course spanning mathematics, physics and computer science. In it, students become familiar with the fundamentals of geometry, how it is used to describe physical motion and how to implement this as an animation.

« 5 » When Geometry of Motion started in 2007, the computer science component was merely a demonstration of computer animations. Most of the time was spent on discussions about how they had been built. We used a public collection of virtual mechanisms (Boytchev, Sendova & Kovatcheva 2011). However, they were standalone programs that were hard to use as learning objects.

« 6 » In 2010 I developed a library called Mecho (Mechanical Objects). Students could use it to construct their own devices (Boytchev 2013a). Since then, new versions of Mecho have been released annually, the last one being completed in March 2014. This version is a result of the research project DFNI-O01/12, financially supported by the Bulgarian Science Fund of the Ministry of Education and Science. The project addresses contemporary programming languages, environments and technologies and their application in the development of software specialists.

« 7 » The design of Mecho follows the major ideas of constructionism. It provides a tool for expressing creativity through

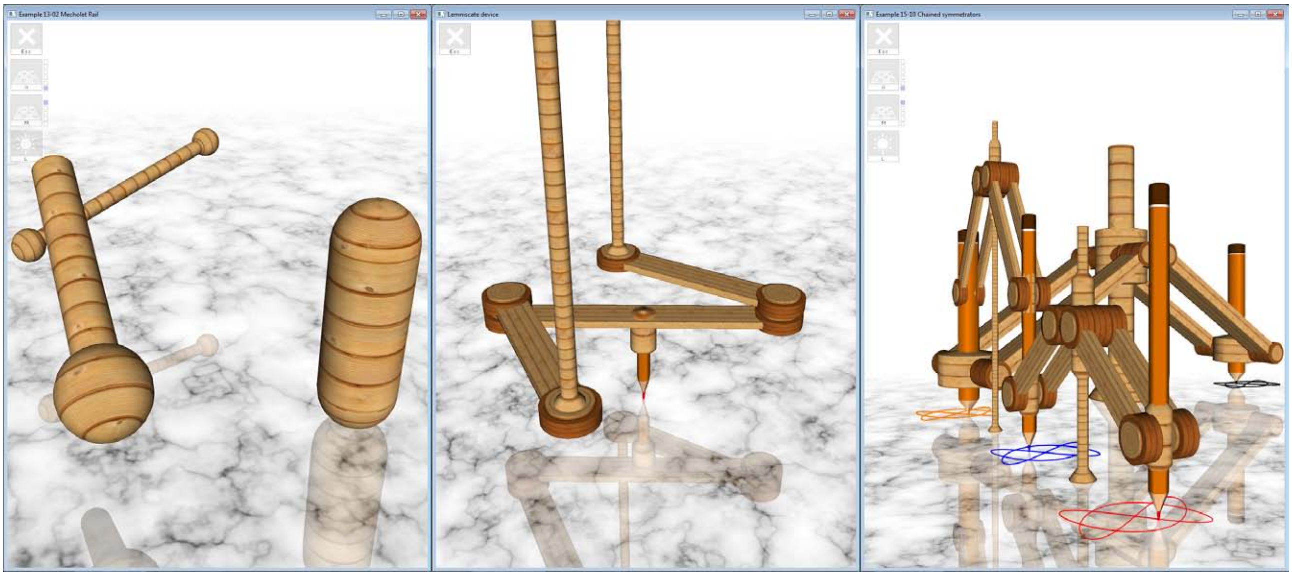

Figure 1 • A virtual component (left), a device (middle) and a machine (right).

public construction of virtual mechanisms. Mecho represents virtual mechanisms as structures with a well-defined hierarchy. There are configurable elements for the basic mechanical components, such as beams and gears. They are used to build simple devices that can be arranged in complex machines. An example of the structural hierarchy is demonstrated in Figure 1. The left image shows individual mechanical components; the middle one displays a virtual device drawing the Lemniscate of Bernoulli; and the last one is a machine exploring chained symmetries.

« 8 » The making of a virtual device is an optional activity for the students. It is up to them to engage in such activity or to ignore it. As a result, very few students have volunteered to build devices. Table 1 shows the number of students for each academic year. During the first three years of Mecho, about 93% of the students avoided it.

Year	Major event	Number of students	
		Enrolled	Working on projects
2007–2008	No projects, just demonstrations	15–20	N/A
2008–2009		15–20	
2009–2010	The course was not offered		
2010–2011	Introduction of Mecho (in Logo)	15–20	3
2011–2012		29	1
2012–2013		36	2
2013–2014	Reimplementation of Mecho in C++	36	20
2014–2015		57	Course is ongoing

Table 1 • Number of students working on Mecho projects.

« 9 » After a detailed analysis of the first three years of Mecho, I identified the key elements that had prevented students from becoming motivated to use it:

- *The learning barrier*: The allocated time for the computer science component of the course was 15 academic hours. This was insufficient to introduce a new programming language (i.e., Logo), to demonstrate motion implementations, to present Mecho and to teach how devices are made. In 2013 I addressed this issue by rewriting Mecho and all teaching materials in C++, a language well-known by the students.
- *The conceptual barrier*: Creating interactive 3D projects is cumbersome, especially if students deal with visualisation and rendering issues. This barrier was resolved by redesigning Mecho so that all activities, such as frame generation and mouse-based navigation, happen "automatically." In this way students focused on the virtual mechanism.
- *The mathematical barrier*: Although students had studied analytical geometry, they still had no practical sense of 3D motion. It was unexpectedly difficult for them to express orientation in 3D space via Euler angles. This

		Compulsory						Procedural						Visual						Hardware						Software						
		0.1	0.2	0.3	0.4	0.5		1.1	1.2	1.3	1.4	1.5		2.1	2.2	2.3	2.4	2.5		3.1	3.2	3.3	3.4	3.5		4.1	4.2	4.3	4.4	4.5		
точки	оценка	proj in time	C::B compat	use mecho	not plag	no 6Lbok	Σ	idea april	proj may	OK file names	PD, CC or GPL	compile OK	Σ	good sizes	5+ elems	has anim	smooth anim	interact buttons	Σ	good joints	mech plausbl	has params	2+ curves	new model	Σ	well format	has // ...	is as class	has docs	resuse in 3.2	Σ	Бонус
16	1.1	1	1	1	1	1	●	1	0	1	1	1	●	1	1	1	1	1	●	1	1	1	0	1	●	1	1	1	0	0	● 3	
0	-0.5	0	0	0	0	0	✗	0	0				✗						✗						✗						✗ 0	
8	0.3	1	1	1	1	1	●	1	0	1	1	1	●	1	1	1	0	0	●	0	0	1	0	0	✗						✗ 0	
1	-0.4	0	0	0	0	1	✗	1	0				✗						✗						✗						✗ 0	
13	0.8	1	1	1	1	1	●	0	0	1	1	1	●	1	1	1	1	0	●	1	1	1	1	0	●	1	0	1	0	0	✗ 2	
13	0.8	1	1	1	1	1	●	0	0	1	1	1	●	1	0	1	1	1	●	1	1	0	1	0	●	1	1	0	1	0	● 3	0
20	1.5	1	1	1	1	1	●	1	1	1	1	1	●	1	1	1	1	1	●	1	1	1	1	1	●	1	1	1	1	0	● 4	1
1	-0.4	0	0	0	0	1	✗	1	0				✗						✗						✗						✗ 0	
13	0.8	1	1	1	1	1	●	0	0	1	1	1	●	1	1	1	1	0	●	1	1	1	0	1	●	1	0	1	0	0	✗ 2	
5	0.0	0	1	1	0	1	✗	0	0	0	0	1	✗	1	1	1	1	0	●	0	0	0	0	0	✗						✗ 0	
1	-0.4	0	0	0	0	1	✗	1	0				✗						✗						✗						✗ 0	
9	0.4	1	1	1	1	1	●	1	0	1	1	1	●	0	1	1	1	0	●	1	0	0	0	1	✗						✗ 0	
1	-0.4	0	0	0	0	1	✗	1	0				✗						✗						✗						✗ 0	
1	-0.4	0	0	0	0	1	✗	0	0				✗						✗						✗						✗ 0	1
17	1.2	1	1	1	1	1	●	0	0	1	1	1	●	1	1	1	1	1	●	1	1	1	1	1	●	1	1	1	0	1	● 4	
2	-0.3	1	1	1	1	1	●	0	0	1	0	1	✗						✗						✗						✗ 0	
2	-0.3	0	0	0	0	1	✗	1	0				✗						✗						✗						✗ 0	1
1	-0.4	0	0	0	0	1	✗	1	0				✗						✗						✗						✗ 0	
13	0.8	1	1	1	1	1	●	0	0	1	1	1	●	1	1	1	1	0	●	1	1	1	0	1	●	1	0	1	0	0	✗ 2	
18	1.3	1	1	1	1	1	●	1	0	1	1	1	●	1	1	1	1	1	●	1	1	1	0	1	●	1	1	1	1	0	● 4	1
20	1.5	1	1	1	1	1	●	1	1	1	1	1	●	1	1	1	1	1	●	1	1	1	1	1	●	1	1	1	1	0	● 4	1

Figure 2 • Score chart with students' progress (rows) across the five levels of criteria (columns); circles mark successfully completed levels.

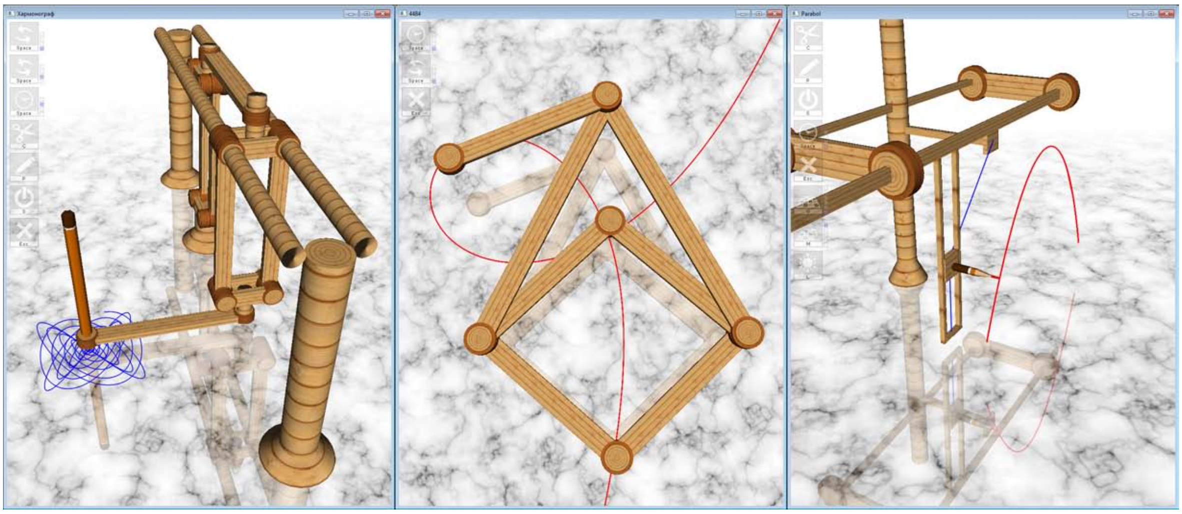

Figure 3 • Virtual mechanisms developed by students.

observation convinced me to exchange mathematical efficiency with user friendliness. I modified Mecho to use comprehensible representations. For example, I implemented 3D orientation by 4 angles instead of the optimal 3 Euler angles.

- *The procedural barrier*: The evaluation of projects considered 25 criteria. Students were introduced upfront to these criteria, but they still experienced problems complying with them. I observed that students were trying to address many criteria at the same time. As a result, they failed to comply with most of them. In 2013 I clustered criteria into five levels and students had to fulfil the levels in a predefined order. The levels represented compulsory criteria, procedural experience, visual experience, hardware experience and software experience (see Figure 2).

« 10 » Our solution for the four barriers made the construction of devices much easier. The number of students almost doubled and their engagement with projects increased eight-fold: from 7% to 56% (see Table 1).

« 11 » The first projects using the redesigned Mecho were delivered in June 2014. Figure 3 shows snapshots of three virtual mechanisms (re-)created by students.

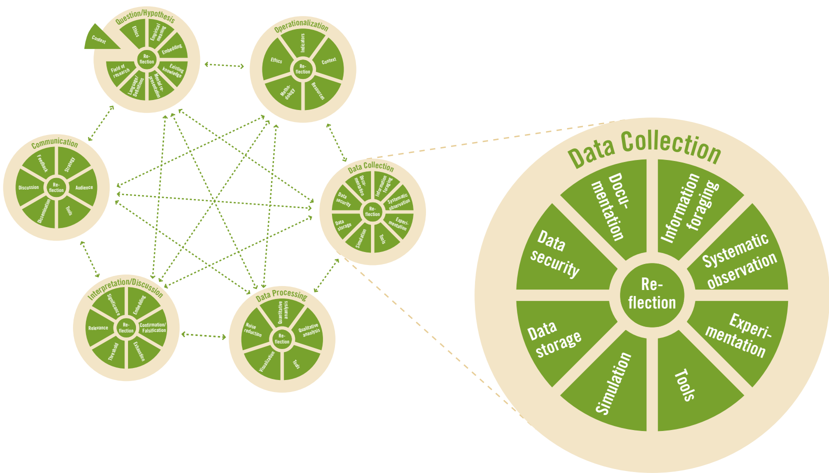

Figure 4 • The weSPOT model of IBL and a close-up of the data collection phase.

Figure 5 • Manual recording of external and internal temperatures.

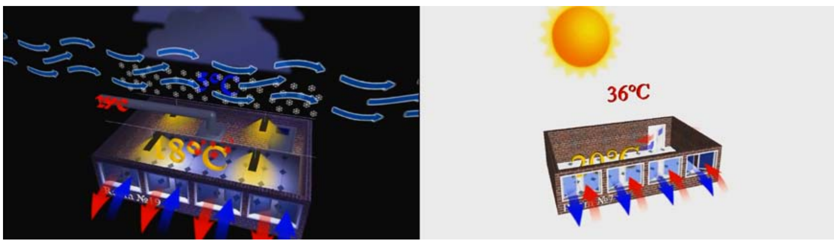

Figure 6 • The classroom simulation software.

Year	Number of students	Grades	Teams
Year 1	60 students (3 classes of 20 students)	Grade 6 (12–13 years old)	Each class was one team
Year 2	60 students (3 classes of 20 students)	Grade 6 (12–13 years old)	Each class was one team, but exploration with the Virtual Classroom was individual or in pairs

Table 2 • Number of students working on energy efficiency pilots of weSPOT.

Constructionism at secondary school level

« 12 » *Inquiry-based learning* (IBL) is a concept closely related to constructionism. In IBL we learn by asking questions and finding answers, rather than by listening to a stream of pre-digested facts. IBL is one of the approaches to implementing constructionism in education and it is recognized as such by institutions at different scales. This section presents software developed for two complementing IBL projects: weSPOT (EC FP7 Programme in Technology Enhanced Learning) and The Role of IT in the Application of IBL in Science Education (Sofia University Science Fund).

« 13 » The goal of weSPOT is to create software tools and know-how for personalization of the IBL environment and management of IBL activities. The project developed a detailed IBL model of six interconnected phases and over 40 components. They are shown in Figure 4 and discussed in Protopsaltis et al. (2014). The hypothesis is that students become researchers and scientists by asking curiosity-driven questions to obtain structured knowledge / context of science concepts. Students are expected to gain skills for effective research, collaboration and creativity.

« 14 » The goal of the second project was to conduct research on the role of IBL in education. It adapted weSPOT's results and focused on science experiments, individualization of education and social collaboration. Several pilot experiments were conducted as a competition between three grade-6 classes. The topic of the competition was "My classroom – The most energy efficient!" The task was to measure temperature variations, weather conditions and the classroom status (such as opened windows, doors, air conditioners, number of people, etc.) Each class produced a report about their measurement including analysis of factors affecting energy consumption. The reports contained suggestions and ideas for reducing the amount of lost energy. The pilot started on 17 November 2012 – the first day of the European Week for Waste Reduction and finished on 5 June 2013 – World Environment Day.

« 15 » During the first year, the students collected data for three months – see Table 2 and Figure 5. More details about this phase of the project are described in Stefanov, Nikolova & Stefanova (2013) and Stefanov et al. (2013).

« 16 » The duration of the data collection was demotivatingly long for 6-graders. It was difficult to keep the students interested in the competition. From the IBL point of view, there were two main problems with this kind of pilot test. It was impossible to repeat the same experiment twice. It was also impossible to change the initial configuration of an experiment and test how this would affect the outcome.

« 17 » For the second year of the project, we decided to provide an alternative approach and developed the Virtual Classroom (Boytchev et al. 2014). This is software containing a non-interactive simulation (see Figure 6) and a standalone interactive 3D application (see Figure 7). The implementation was based on decisions that were initially considered risky.

« 18 » The first risky decision was to make *a continuous simulation*. There were no means to start or stop the virtual classroom. It was running even when the students were setting the parameters of their experiments. The second risky decision was to make *unrestricted simulation*. This is the ability to set unnatural initial conditions, such as snowing at 40°C. In such a case, the air temperature would smoothly go down until the physical model reached equilibrium. The main feature of the simulation mechanisms was that it only managed transfer of energy in small quantities towards equilibrium.

« 19 » Continuous and unrestricted simulation contributed to a better simulation, closer to the actual world, where students cannot control the fabrics of observed phenomena.

« 20 » To support the inquiry process, the Virtual Classroom was distributed *without any documentation*. Thus, students and teachers had to find by themselves all the software's features: from navigation to conducting experiments. There was no description of the simulation mechanism. For example, students conducted experiments to find whether the number of people in the classroom affects the air temperature.

« 21 » The pilots with the 6-graders were video-recorded, and snapshots of the recordings are presented in Figure 8. The analysis of the recordings showed that the software provoked inquiry learning and active constructionism. Every student worked at his or her own pace while gaining scientific skill.

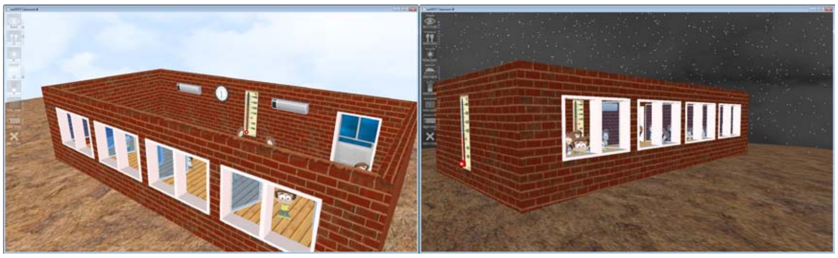

Figure 7 • The interactive virtual classroom.

Figure 8 • Students exploring the Virtual Classroom.

« 22 » The pilots were conducted in the spirit of constructionism. Students learned by constructing public entities. The process of construction was also public. While observing the progress of their classmates the students soon started to exchange ideas. One interesting and unplanned observation was how students experienced the scientific importance of details. Several students conducted "equivalent" experiments, but got opposite results because of subtle differences in the initial conditions. This experience was quite valuable. It helped gain the skill of distinguishing important from unimportant factors.

Deconstructionism in education

Phases of constructing knowledge

« 23 » The experience with university and secondary school students showed that it is not straightforward to utilize constructionism. Although we created different tools to support this application, students still experienced problems.

« 24 » The process of learning through construction can be split in two phases – deconstruction and construction, shown in Figure 9. I use the word *deconstruction* in the sense of decomposing or breaking down something into reusable entities. In contrast, the meaning of *destruction* would be to destroy something. The left image in Figure 9

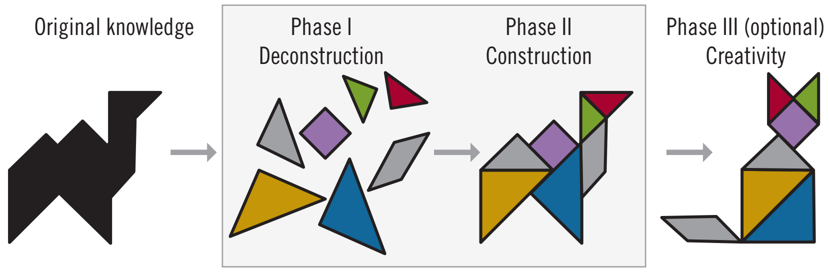

Figure 9 • Phases of learning through deconstruction and construction.

represents some knowledge. The first phase of learning is to decompose this knowledge into smaller yet meaningful entities for the learner. These entities are used as building blocks to construct the personal knowledge, which is not necessarily the same as the original knowledge. There is a third phase where new knowledge is created by rearranging the entities in another way.

« 25 » Most of the literature about constructionism is focused on Phase II – the construction. Many approaches have been developed in order to ease this phase. ICT solutions implemented by providers of educational content also focus on construction.

« 26 » The experience with students at Sofia University and secondary school students showed that the most difficult phases are I and III. The creativity phase is usually optional. There is no universal algorithm on how to create creativity. Most of the conventional lessons are designed to reach up to Phase II. However, Phase I is not optional. It is a prerequisite to the construction phase. Any failure to deconstruct knowledge leads to failure in Phase II. Even more, the skills required for effective deconstruction are comparable to the skills required for creative activities.

« 27 » Table 3 presents the entities of the original knowledge in the discussed cases. The original and the constructed knowledge share the same cells because it is expected that they are the same. The table also lists some of the skills found to support effective deconstruction.

« 28 » Generally, finding deconstruction entities is difficult. Fortunately, in the pre-set educational environment of our pilot cases it was not so hard to identify them. The actual difficulty of deconstruction in education is not the elements in each phase, but the transition between phases.

« 29 » The last row of Table 3 contains ideas of possible creative artefacts. Some of them were partly realized by the students. For example, the left-most mechanism in Figure 3 differs from the traditional implementations of harmonographs, and was invented by a student.

« 30 » The issues with the deconstruction phase were identified a long time ago. Resnick (1990) describes what he calls *problem-decomposition bugs*, which point to the difficulties of decomposing problems into simpler entities. This problem decomposition is presented as a space of two dimensions: *functional decomposition* and *agency decomposition*.

Manifestation of deconstruction

« 31 » The nature of the deconstruction phase is elusive and vague. It happens behind the scenes. It is often interlaced with activities from the construction phase. This makes it difficult to identify the activities that occur during deconstruction. The experience with university and secondary school students shows that creative deconstruction has many distinct manifestations. Some of the most commonly observed ones are: debugging, animation design, problem solving and pattern recognition.

Debugging

« 32 » From a deconstructionismic viewpoint, debugging is the process of decomposing a running program into entities that help us eliminate its malfunction. There are tools that ease this process, but they are often insufficient for effective debugging. These tools are good for tracking the expression of a bug, but not the actual bug, which may be located in a completely different area in the code. When people debug, they build a mental interconnected and dynamic representation of the functional components of a program, the data flows and the logic in each step of the execution process. Debugging tools still cannot automatically extract this representation and present it in a comprehensible way.

Animation design

« 33 » When creating an animation, students face the problem of representing a motion as a composition of simpler motions based on mathematical functions. This deconstruction is hard for many students. They lack the skills to see (or to imagine) how a composite animation could be represented as an outcome of fundamental functions.

« 34 » During the Geometry of Motion course, the students observe various models of physical motions. One of the most difficult steps is to approximate these motions with a limited set of mathematical functions. An example from Lecture 10 is a pair of cubes bouncing off a vibrating spinning disk, as seen from a viewpoint orbiting the whole scene. All motions in this example are implemented with $sin(x)$.

Problem solving

« 35 » Problem solving requires understanding the problem and its decomposition into entities used to compose a solution. For mathematical problems, these entities could be theorems and lemmas, but they could also be algorithms. In the case of the course Geometry of Motion course, the deconstruction phase contains activities for inventing how to represent a given motion as a mechanical linkage. This is traditionally far more difficult than the construction phase, which is when the virtual mechanism is being built following an existing design.

Pattern recognition

« 36 » This is the ability to identify meaningful entities in an otherwise chaotic-appearing texture. Patterns are not only visual. They could also be patterns of algorithms, patterns of methodology, patterns of approaches and patterns of behaviour. A proper identification of patterns contributes to successful construction. The manifestation of deconstruction is exactly the process of finding the pattern, i.e., the texture is decomposed into meaningful entities that exhibit the nature of the pattern.

« 37 » The main problem of deconstruction is that it is hard to formalize the deconstruction phase. As a consequence, it is hard to provide methodological, pedagogical and technological tools that support it. The deconstruction phase is almost completely confined to being realized by the students themselves.

Deconstruction outside education

« 38 » In "Phases of constructing knowledge," I defined deconstruction as the process of decomposing or breaking down something into reusable entities. Because this definition is intentionally all-embracing, it is applicable to things outside education. The rest of this section contains ideas and personal observations on one possible way to map the notion of deconstruction onto a more general context.

« 39 » People are prone to deconstruction. It is not an artificial activity introduced through and for learning only. Traces of deconstruction can be observed in many situations beyond the traditional educational context. Deconstruction is actually a part of our lives. All the following examples are deconstructions:

- A child breaking a favourite toy just from curiosity to see what is inside
- A person trying to distinguish the ingredients of a meal by the aroma of its spices
- A scientist reverse-engineering a biological mechanism.

« 40 » Apparently, deconstruction is not just something that happens sporadically through our lives. It is also a major scientific arsenal. For example, the deconstruction of mathematics leads to the "invention" of its fundamental axioms. It is not a one-way route. Different mathematical sciences, especially the geometries, have been successfully deconstructed into different sets of axioms. Once we have the axioms, we can construct back the corresponding mathematic.

« 41 » With a more global scope, the understanding of nature goes first through its deconstruction into sciences such as physics, biology, chemistry and astronomy. This deconstruction phase is vital. Without it we will be overwhelmed by the complexity of nature. The construction phase is already happening. It is the reverse process of merging back different sciences and building multidisciplinary relations, such as astrobiology and medical informatics. The construction phase will end when all sciences merge coherently into one.

Constructionism and deconstructionism

« 42 » I define *deconstructionism* as a distinct perspective on the same objects and processes that are requisites for the constructionism. Constructionism is focused on the personal construction of ideas and relations through the construction of real-

	University experience with Mecho	Secondary school experience with Virtual Classroom
Knowledge to be deconstructed and then constructed	▪ Expressing complex motion with a limited set of functions ▪ Mapping mechanical linkage to a geometrical curve and vice versa ▪ Transforming abstract mathematical linkage to a physically possible linkage	▪ Finding questions leading to scientific approaches in finding answers ▪ Conducting experiments in a dynamic environment ▪ Studying the behaviour of unknown complex systems
Skills supporting deconstruction	▪ Recognition of graphs of functions ▪ Decomposition of composite functions into fundamental functions ▪ Approximation of functions via simpler functions ▪ Solutions to mechanical collisions ▪ Reversing kinematic ▪ Expressing motion with higher degree of freedom as a composition of several lower degree motions	▪ Observation of real-life and simulated phenomena ▪ Extracting optimal set of parameters capturing observed behaviour ▪ Relating environmental changes to system configuration and vice versa ▪ Eliminating false positives and false negatives ▪ Conducting different experiments to verify a single hypotheses
Entities created by deconstruction	▪ Fundamental mathematical functions (e.g., *sine* and *absolute value*) ▪ Mathematical operations (e.g., *vector addition*, *linear combination*) ▪ Basic mechanical components (e.g., *beams*, *rails*, *gears*)	▪ External factors affecting energy consumption (e.g., *temperature*, *clouds*) ▪ Internal factors affecting energy consumption (e.g., *door*, *air conditioning*) ▪ Mathematical relations (e.g., *energy consumption vs temperature difference*)
Possible artefacts of creativity	▪ A mechanism that is functionally equivalent to existing ones but uses fewer mechanical parts ▪ Know-how about mechanisms with motion that can be expressed by 3rd degree polynomial functions	▪ An algorithm for smart control of energy consumption in buildings, based on precise sensing and forecast of weather ▪ A simple formula for quick approximation of power requirements in a building

Table 3 • Knowledge, skills and deconstruction entities in the university and secondary school experiences.

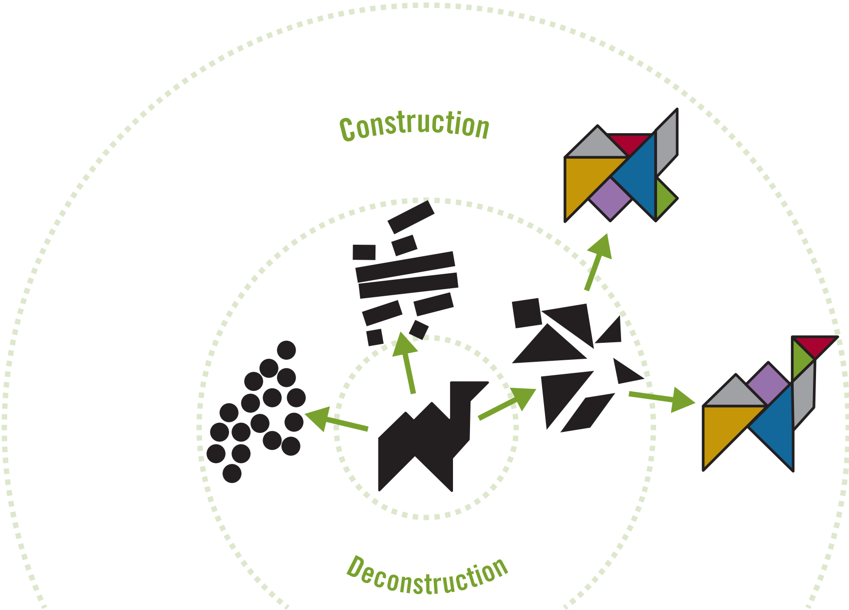

Figure 10 • Lack of determinism in both deconstruction and construction.

life artefacts. Deconstructionism is focused on the personal understanding of ideas and relations through the public deconstruction of real-life artefacts.

« 43 » Similarly to the relation between constructivism and constructionism, *deconstructivism* is about the mental private decomposition of ideas and relations, while *deconstructionism* is about the deconstruction of a tangible artefact or about the public deconstruction of a concept. In this sense, the sequence of phases as pictured in Figure 9 is not to be considered sequential or linear. In fact, the deconstruction phase is repeated several times until the initial knowledge is decomposed into proper ingredients that can be used for reconstruction of the personal knowledge. This may take several attempts, as indicated in Figure 10.

« 44 » The deconstruction phase is not deterministic, just as the construction phase is not deterministic. Problems in learning due to excess cognitive load or a cognitive barrier occur predominantly in the deconstruction phase. When students cannot relate a new concept to their previous knowledge, they actually fail to decompose that new knowledge and get stuck. This is the focus of the deconstructionism.

The future of education

Deconstruction of education

« 45 » In an interview for *New Scientist*, Noam Chomsky says "If you're teaching today what you were teaching five years ago, either the field is dead or you are" (Lawton 2012). Although he is referring to linguistics, the same applies for all domains in education. The digital era is having a tremendous impact on how we learn. People have already created digital content that exceeds the capacity of available storage. Modern technology challenges the traditional pillars of the educational model: student, teacher, textbook and school. The average digital weight (volume of created digital content) of a student is overtaking the digital weights of the teacher and the textbook combined. The accessibility of digital content is displacing the school as a main source of knowledge. It questions the very nature of the traditional school and textbook.

« 46 » For centuries, education was changing incrementally. Today it cannot cope with the exponential development of technology. There are two possible paths: either education distances itself from technology, or embraces it. Clearly, technology and education cannot be separated. The attempts to restore the balance between them by mere reconstruction of education do not produce sustainable results. The introduction of ICT in the classroom is unable to synchronize education and technology. An alternative approach is to *deconstruct education* into its fundamental components, then to build a conceptually new education. Thus, deconstructionism could become the major player in reshaping how people teach and learn. Deconstructionism *in* education is hard to achieve. Deconstructionism *of* education will be much harder.

Factors to consider

Digitality

« 47 » Several factors may affect the future of education and *digitality* is one of them. The word *digital* has two meanings related to education. The first one is *finger*. It is what education was for centuries – learning by hands-on activities. The other meaning is related to *numbers*. It is what education is trying to become nowadays – learning by manipulation of virtual entities. Unfortunately, the numerically digital education is becoming dominant and is dislodging some of the best practices in the "fingerly" digital education (Boytchev 2013b). Fortunately, the advances in technology make it possible to merge both digital educations.

« 48 » The pilots discussed in this article could be significantly enriched by this digitality. For example, a future version of the Virtual Classroom may provide a tactile interface to the simulation, while a future version of Mecho may communicate with a 3D printer to produce tangible mechanisms. Thus digitality will allow the students to convert their virtual artefacts into tangible artefacts.

Ubiquity

« 49 » There is a trend of promoting ubiquitous learning (*u-learning*). This is a learning that "enables anyone to learn at anyplace at any time" (Yahya, Ahmad & Jalil 2010: 117). Ubiquity in future education will develop in several aspects. First, ubiquitous learning will span not only over space and time, but through any media. Learning will happen in parallel through a variety of media including the social media. Thus

PAVEL BOYTCHEV
is an associated professor and researcher at the Faculty of Mathematics and Informatics, Sofia University. His research interests are in the areas of developing courses, educational software, authoring software, computer graphics and animation, visualization, multimedia, and design and implementation of programming languages. He has created numerous educational applications and educational programming languages. He is an author of a dozen courses, hundreds of computer-generated video clips and more than a thousand computer demo programs in his areas of interest.

students will have their own imprint on the learning process. Second, teaching will also become ubiquitous. The relation between *u-teaching* and u-learning is as the relation between deconstructionism and constructionism. The main goal of u-teaching is the decomposition of learning content that renders it u-learnable – this is a challenge with yet unknown complexity.

« 50 » A possible impact of ubiquity on the Virtual Classroom and Mecho is to allow students to play with the software at anytime and anywhere. There are already plans for newer versions of the software based on mobile 3D graphics. This will make the Virtual Classroom and Mecho mobile-friendly and platform-independent.

Transparency

« 51 » Modern technology is getting more transparent and less obtrusive. Much technological and educational power is encapsulated in small yet smart devices. The advance in technology is shifting learning to a new course. I expect that future learners will not learn mathematics, but will experience it. The current model of education creates an image of the world through which people learn. In a technologically transparent future education, people will learn directly from the world around them using all their senses. First attempts in this direction have already been made by the research on virtual, augmented and immersive realities.

« 52 » It is hard to imagine what the virtual classroom would look like in a future of immersive technologies. Most likely, the virtual and the actual classroom will be indistinguishable, or even the same. Because of digitality, ubiquity and transparency, the concept of classroom may become void.

Conclusion

« 53 » Constructionism approaches are applied to education with variable degrees of success. This article describes the application of constructionism to university and secondary school levels. One of the cases is a new course, where students construct virtual mechanisms exhibiting or representing mathematical properties. The other case is of interactive software for inquiry-based learning. This software allows students to conduct experiments in a simulated microworld, to collect data for raising or proving hypotheses and to investigate unknown relation between entities. Although both cases provide an interesting and motivating medium fostering education in a constructionistic way, there is one specific phenomenon that emerges from every pilot case. It is the phase of deconstruction, which is routinely neglected. A possible reason is the inherent difficulty of the deconstruction. This makes it as hard to achieve as it is to teach creativity.

« 54 » Deconstruction is an important aspect of science and education. Yet, there are no methodological, pedagogical and technological tools that support constructive deconstruction. Education has been incremental for centuries. It cannot cope with the exponential growth of technology and is falling behind. There are efforts to shape the education of the future, including utilizing constructivist and constructionist approaches. However, I advocate that to be able to construct a completely new and adequate education, two steps are needed upfront: (1) acknowledging and supporting deconstruction in education; and (2) the deconstruction of education itself. When these two steps are completed, it will be possible to construct a new type of education. Meanwhile special attention must be paid to three factors: the symbiosis of the two digital educations, ubiquity of learning and teaching and the increasing transparency of technology.

Acknowledgements

The work described in this article is financially supported by: project DFNI-O01/12 Contemporary Programming Languages, Environments and Technologies and their Application in Development of Software Specialists, supported by the Bulgarian Science Fund; project FP7 318499 weSPOT, supported by the EC ICT FP7; project 108/19.04.2013 The Role of IT in the Application of IBL in Science Education and project BG051PO001-3.3.06-0052 Forming a New Generation of Researchers by Supporting PhD, Post-PhD and Young Scientists, supported by the Sofia University Science Fund.

Received: 20 February 2015
Accepted: 21 April 2015

Open Peer Commentaries

on Pavel Boytchev's "Constructionism and Deconstructionism"

Deconstruction in Software Construction

Gerald Futschek
Vienna University of Technology, Austria
gerald.futschek/at/tuwien.ac.at

> Upshot • Boytchev's deconstructionism looks at first glance like a game of words. Upon a deeper view of the subject, he focuses our attention on the importance of deconstruction to the construction process, which is highly connected to creativity. In my contribution, I want to point out the close relationship of Boytchev's deconstruction to the software development process, where requirements analysis corresponds to deconstruction and software design and implementation correspond to construction. Creativity is an important asset in any kind of software development, where life-long learning is essential.

« 1 » At first glance, the article's title was provocative to me. I felt that deconstruction, as the opposite of construction, means something like destruction. How can destruction help to construct knowledge? I was aware that destruction is the contrary of construction and that destruction is sometimes a necessary prerequisite of construction. Shiva, one of the major deities of Hinduism, is known not only as destroyer but also as creator. Often the need for a construction arises from a destruction. So a close relationship between construction and destruction exists.

« 2 » Pavel Boytchev gives a wise definition of deconstruction: a decomposition of something into reusable entities (§24). These reusable entities can be used in the following construction phase. So deconstruction is not the opposite of construction, it is an essential part of it and incorporates a great deal of necessary creativity. Boytchev's definition of deconstruction is therefore completely different to what can be expected from the title.

« 3 » And why create a new –ism out of deconstruction (§42)? Here, I think it is not justified to define deconstructionism and use it in the same line as constructionism. It is, from my point of view, really a game of words. Deconstruction is, by definition, part of the construction process and therefore subordinate to construction.

« 4 » Boytchev's description of the phases of learning through deconstruction and construction strongly reminds me of core activities of software development processes: requirements analysis, software design and software construction. For more details on software development processes, see, for example, the recent book by Murali Chemuturi (2012). When a larger piece of software has to be developed, first of all the problem domain has to be understood by the developers. The result of analysing the requirements is a structured model of the problem domain. This is usually a decomposition of the problem domain into smaller entities that interact with each other. The main purpose of breaking it up into its component parts is not only the better understanding of the problem domain but also the reuse of the parts in the constructive activities of software design and software construction. In the software design activity, a plan for the construction of the software is created. A part of this plan results from the components of the requirement analysis.

« 5 » Creativity is an essential property of software development. For each problem, there are endless potential software solutions that tackle the problem. It is a creative act to find a better solution that is highly-appreciated by users. Not all steps of the software development process can be performed just by following well-defined rules. The experience and creativity of the software engineer strongly influences the outcome of his activities. The way we look at the problem domain and how we deconstruct it into manageable parts strongly influences the quality of the final software product.

« 6 » Why do I connect constructionist learning with software engineering? It is not only the similarity of the processes. It is also the fact that there is constructionist learning in the software industry. It is a life-long learning process that forms good software engineers. They learn from their own and other's drawbacks and successes. So the software engineering activities can be seen as a life-long constructionist learning process.

« 7 » It is not only the future of learning that is massively influenced by the exponential growth in technology (§46), constructionism may also positively influence information technology with its potential in the life-long learning process of software engineers.

« 8 » The two practical examples of constructionist learning that are presented at the beginning of Boytchev's article show very impressively how technology may stimulate the learner's creativity. These technologies are very powerful; they allow a large variety of learning paths and so make room for creativity. The library Mecho (§6)is a very flexible programmable tool that allows university students to create wonderful new virtual mechanisms. Completely different is the tool that was designed for school students to explore a virtual classroom (§17). The students can also provoke very extreme weather situations and study their effect on the virtual classroom. Furthermore, this simulation tool allows a variety of inquiry-based learning experiences that support the creative thinking of the learners.

Gerald Futschek is professor at the Institute of Software Technology and Interactive Systems at the Vienna University of Technology and chair of the special interest group on teacher education of the Austrian Computer Society. His areas of interest include software engineering, program verification and informatics didactics. Homepage: http://www.ifs.tuwien.ac.at/futschek

Received: 15 June 2015
Accepted: xxxx

Construction and Deconstruction

Brian Harvey
University of California, Berkeley, USA • bh/at/cs.berkeley.edu

> Upshot • Pavel Boytchev's article calls attention to the fruitful dialectic between building things and taking them apart: No successful construction without deconstruction. Of course by using the word "deconstruction," he is also implicitly invoking the critical-theory sense of the term, inviting us to deconstruct constructionism. I found the article fascinating on both levels.

« 1 » This is a very provocative target article. Its central thesis, that deconstruction is essential to successful construction, is an idea that, once stated, is obvious, but it is far out of the mainstream of constructionist vocabulary. I think, though, that the *practice* of deconstruction may be more prevalent than its analysis.

« 2 » For one thing, it is a commonplace observation that you learn how to build things by looking at, and looking inside, already-built examples. Sometimes this involves physical deconstruction, as in Pavel Boytchev's example of the child deliberately breaking a toy to see how it works. But I am thinking also of the "Look inside" button on Scratch project pages. The visitor's first view of someone else's Scratch project focuses on its behavior and purpose. But by clicking "Look inside," the visitor can examine the Scratch program that makes the project function, and can even modify or "remix" it.

« 3 » Secondly, people try to build learning environments based on a small number of simple parts. The canonical example is Lego, which has been much more successful than the Erector sets of my own childhood – in which the main components were metal bars of many different lengths that could be bolted together, along with specialized connectors such as angle brackets – in part because of the wide range of projects that can be built out of (many copies of) a single part, the Lego brick. Such environments come pre-deconstructed. The same could be said of the effort in the design of Scratch to minimize block shapes and to minimize the number of primitive blocks altogether.

« 4 » Thirdly, the emphasis on physical stuff in the Maker Movement makes the component parts much more readily visible than in either computer software or even more abstract mathematics.

« 5 » I would be interested in a discussion of how the idea of deconstruction fits in with the (I think) overlapping idea of demystification. One way to demystify a black box is to see how it is built. But there are also demystification strategies that, for example, try to shed light on people's motivation, or on sociological ideas such as "institution." Is the purpose of deconstruction demystification, or something else?

« 6 » I think some of the points in the article are made in too much of a hurry. Here are two examples.

« 7 » Debugging as deconstruction (§32): I understand the overall point, I think; in order to debug you have to think analytically about the pieces of a program. And I get that current debugging technology is weak. But I do not think I quite got how a deconstructionist perspective can lead to better debugging tools. This could be a research project in itself, and the one paragraph here leaves me dissatisfied.

« 8 » Technology and education (§45): There is a claim here that very rapid change in technology makes it hard for education to keep up. I see how that would apply for education about technology – which is sort of the point of the Noam Chomsky quotation in §45. And I can see how education might not be making the best possible use of technology: although, for example, schools have been quick to jump on the tablet bandwagon, as far as I know they have not yet found any purpose for which tablets are better than laptops. (I do get that they are cheaper, although now you can get netbooks instead.) But children are not so very different in how they construct knowledge, are they? If that is the intended claim, it needs much more discussion. Otherwise the point about exponential change is a little glib.

« 9 » In contrast, the discussion of the virtual classroom project is detailed, vivid, and very convincing. Figures 9 and 10 are a wonderful encapsulation of the central point. I would have been glad if the last few pages of the article were more a laying out of a specific program for future research, rather than a collection of brief mentions of largely unsupported ideas.

« 10 » I do wonder why students of the virtual classroom were required to deconstruct not only the physical model being simulated but also the simulator as a piece of software. Why, for example, make a point of not documenting which keystroke does what in the user interface? Does *that* deconstruction task not just distract from the deconstruction of the classroom environment?

« 11 » There is another paper to be written about the deconstruction of education. Does that mean something more than the critique of education that starts (maybe) with Jean Jacques Rousseau and goes through John Dewey to Paulo Freire and Ivan Illich? Is there some way in which computers change this critique? What *are* the component parts of education? Learner, teacher, and content? Maybe plus the learning community? Or does Boytchev mean something more detailed than that? When I visit a school, I generally find myself thinking first not about technology, but about the social relations in the classroom – mostly the one between the teacher and the students, but also relations among the students.

Brian Harvey is Teaching Professor Emeritus in Computer Science at the University of California, Berkeley. He has taught every age from eight to adult, teaching programming to the kids and the teaching of programming to the adults. He is the author of the three-volume *Computer Science Logo Style* books for teenagers. His recent efforts have been in co-developing the Snap*!* visual programming language (with Jens Mönig) and the "Beauty and Joy of Computing" curriculum (with Daniel Garcia).

Received: 29 May 2015
Accepted: 22 June 2015

"Deconstructionism" – A Neglected Stage in the Constructivist Learning Process?

Wayne Holmes
UCL Institute of Education, University College London, UK
w.holmes/at/ioe.ac.uk

> Upshot • Boytchev identifies "deconstruction" as a neglected but essential stage in the constructivist learning process. Drawing on two studies, one in a university and one in a secondary school, for which software was designed to facilitate constructionist student learning, the author argues that the first phase of learning is the decomposition of knowledge into smaller yet meaningful and reusable entities, which are used as building blocks to construct both personal and new knowledge.

« 1 » Pavel Boytchev's target article, "Constructionism and Deconstructionism," encompasses two studies from which is inferred a novel theoretical perspective. Both elements have their merits. While the studies highlight benefits and difficulties of implementing constructionist learning environments, the exposition of "deconstructionism" highlights a neglected stage in the constructivist learning process. On the other hand, the article makes no reference to two highly pertinent sets of ideas, to either a design-based approach that may have informed both the educational technologies and the research studies or to the earliest theory of deconstructionism, by the French philosopher Jacques Derrida.

« 2 » Boytchev's first study (§§4–11) focuses on the use by university students of a tool, known as Mecho, designed to facilitate the construction of virtual mechanisms by combining basic mechanical components to build simple devices that in turn can be assembled to make complex machines. However, while the software is exciting and appears to have huge potential, the discussion is somewhat superficial. Nevertheless, an especially interesting point emerges.

« 3 » The researchers encountered enormous difficulty encouraging students to use the system to build devices, only around 7% of the students did so in the first year, which the author ascribes to four key "barriers": insufficient time to learn the new programming language, which was addressed by rewriting the code in a language already familiar to the students; the software interface predominating over its function, addressed by automating much of the interface and its tools; the complexity of Euler-based 3D mathematics, addressed by prioritising ease-of-use over mathematical fidelity; and complex learning criteria, addressed by introducing a predefined structure of learning criteria clusters.

« 4 » The author reports that these "solutions" led to an 8-fold increase in the number of projects completed, but provides no evidence to support such a conclusion. While it does seem likely that making the tool fundamentally easier to use did help those students who chose to explore the software, many other factors may also have had an impact on whether or not they did: factors such as how the software was presented by the instructor, how projects contributed to course credits, and issues of student task comprehension (what exactly are the students supposed to do?) associated with many constructivist learning environments.

« 5 » Boytchev's second study (§§12–22) focused on an inquiry-based learning project in a secondary school using the Virtual Classroom software designed to help investigate classroom energy use in order to generate ideas for reducing energy wastage. Curiously, the software was "distributed *without any documentation*," which meant that the students had to find by themselves all the navigation and experimentation features. However, while the author claims that "the software provoked inquiry learning and active constructionism," again we are provided with little evidence. Indeed, it is unclear how the students engaged with the software, what they were attempting to achieve or how they went about doing so. This is especially pertinent when claims are being made about constructivist (or constructionist) learning environments, which are known to be unsuccessful if the learner fails to come into contact with the relevant information or if, having done so, fails to make the necessary connections (Kirschner, Sweller & Clark 2006). The issue is that without some guidance, constructivist learning environments "may fail to promote the first cognitive process, namely, selecting relevant incoming information" (Mayer 2004: 17). Even Jerome Bruner (1961) counselled that discovery cannot be made *a priori* or without at least some guiding theory or knowledge of the domain.

« 6 » The development and research of the Virtual Classroom may also have benefited from the adoption of the iterative and theory-generating approach of design-based research (Design-Based Research Collective 2003). Rather than designing a complete tool and then simply refining it when its limitations become clear, more might have been achieved from an on-going iterative approach, with an initial prototype grounded in theory that is researched in an authentic context, the outcomes of which are used to generate both a second iteration of the tool and local theory – all of which together constitute a process of design, intervention, analysis and reflection that is repeated until robust conclusions may be drawn.

« 7 » Before addressing Boytchev's "deconstructionism," some reference has to be made to Derrida's "deconstructionism," a critical approach to understanding cultural artefacts that was introduced in the 1960s and predominant in a various diluted forms in much academic discourse in the 1970s and 1980s: "deconstruction is not an operation that supervenes afterwards, from the outside, one fine day. It is always already at work in the work" (Derrida 1989: 73). The point here is not that Boytchev should accommodate Derrida's deconstructionism but that any decision to appropriate such a heavily loaded *word* should make some critical reference to the original *concept*, if only to contextualise and distinguish the new proposed usage.

« 8 » Nevertheless, Boytchev's "deconstructionism" does highlight a neglected stage in the constructivist learning process and is worth some consideration. For Boytchev, the "first phase of learning" is to deconstruct / decompose / break down "some knowledge" into smaller yet meaningful-for-the-learner and reusable entities (§24), which are used as building blocks to construct personal knowledge, not necessarily the same as the original, and can also be used to create *new* knowledge. However, as is all too common in much academic writing about educational technology, this argument is at least partly based on conceptions of "knowledge" and "learning" that might themselves benefit from a process of critical "deconstruction." Further, is deconstruction by the learner always necessary for learning? Is it not possible that learners might encounter previously deconstructed elements of knowledge, deconstructed for them by their teacher or embodied in well-designed learning environments by learning designers, that they explore and – with guidance appropriate for the individual learner – construct into personal or new knowledge?

« 9 » Boytchev focuses on deconstruction by the learner, what is described as an "elusive and vague" phase that is both necessary for and often interlaced with the construction phase. In a table, the author summarises the original knowledge, skills and deconstruction entities that have been drawn from the two studies, which are then used to bolster the argument. While the discussion of the deconstruction phase has some logical coherence and suggests much potential, more detail about the studies would have made it easier for the reader to make similar connections.

Wayne Holmes is currently a post-doctoral researcher at the London Knowledge Lab (UCL Institute of Education, University College London), where he is working on the EU funded iTalk2Learn project. He also currently teaches on the MSc Education, Technology and Society at the Graduate School of Education, University of Bristol. He received a DPhil in Education (Learning and Technology) from the University of Oxford, and his research interests are on the application of the learning sciences, psychology and neuroscience, to learning and pedagogy.

Received: 8 June 2015
Accepted: 22 June 2015

Author's Response: Does Understanding Deconstruction Require Its Deconstruction?

Pavel Boytchev

> Upshot • I describe my perception of deconstruction, including the controversial point of view that deconstruction is actually construction. I also provide more details about the some of the design decisions in the software, and how these affected the students' experience.

« 1 » My target article was deliberately iconoclastic, and this was also expressed by **Gerald Futschek** (§1) and **Brian Harvey** (§1). The article is not to be considered as a final research piece. Instead, it is more like a wake-up call, trying to throw readers off-balance. I did not give answers, I only raised questions.

« 2 » In the summer of 2013, I reached the idea of deconstruction as a process that is essentially part of construction. This was publicly announced at Constructionism 2014 in Vienna. A long time before that, I had a vague idea about something mysterious that happens during learning. It became less vague to me when I named it "deconstruction." The reason for picking exactly this name was to capture the essential idea and importance of this process. I found that "deconstruction" was already loaded with some meaning. As **Wayne Holmes** (§1, §7) mentions, Jacques Derrida introduced "deconstruction" some 50 years ago. Yet, his deconstruction does not map sufficiently with my idea. To minimize misunderstanding, I gave a specific definition of what I mean by "deconstruction" in §24 and again in §38; and peer comment authors have acknowledged this definition (**Futschek** §2; **Holmes** §8).

The constructive nature of deconstruction

« 3 » In §26 and §31, I discussed briefly the creative nature of deconstruction. Why did I think it so? If construction is creative, why should deconstruction be creative too? Changing the perspective, one can see that deconstruction is a process of construction of small entities. This construction is non-deterministic (§42) and creative.

« 4 » This raises the question of whether it is possible to support deconstruction. **Harvey** (§§2f) drew parallels with Lego bricks, an Erector set and Scratch, while **Holmes** (§8) described "previously deconstructed elements of knowledge," prepared and deconstructed for the students. Indeed, these parallels visualize nicely that the direct result of deconstruction is a set of simple entities used to (re)create knowledge. It may appear that such pre-deconstructed sets would ease (de)construction, and this might be correct. However, I see a disadvantage. The process of decomposing into basic "bricks" is an essential process: what Lego, Erector, Scratch, Logo or any previously deconstructed element does is to bypass this process, effectively nullifying its eventual educational impact. I consider deconstruction to be a creative process, and different people may want to deconstruct the same thing into different sets of elements.

« 5 » Here is a personal example. In my early years I studied how to add numbers. There was an algorithm for how to do this, simple and effective, but I used my own algorithm. For example, I would calculate 85 + 37 by transferring quantities from 37 to 85 in a way that zeroes its least significant digits one by one:

$$85+37=(85+5)+(37-5)=90+32=$$
$$(90+10)+(32-10)=100+22=122$$

I could do the opposite too:

$$85+37=(85-3)+(37+3)=82+40=$$
$$(82-60)+(40+60)=22+100=122$$

Forty years later, I still use this approach, not the one in the textbook.

« 6 » In §5, Harvey raises an interesting question about demystification. According to me, the relation between deconstruction and demystification is that demystification is just an interim subprocess during deconstruction. Demystification does not encompass the idea of subsequent (re)creation and its goal is not to create reusable entities.

« 7 » I understand that the topic of deconstruction and deconstructionism needs far more attention and detail. It was impossible to discuss all the issues in sufficient depth in a single article. So I agree with the observations of Holmes (§2, §9) and Harvey (§§6f). I also agree with the suggestion that "there is another paper to be written about the deconstruction of education" (Harvey, §11).

The software environments

« 8 » I based our discussion about deconstructionism on the software environments developed for and used by research projects. These projects had their own goals, plans and results. I mention this just to make it clear that the purpose of Mecho and Virtual Classroom was not to investigate deconstruction. However, I found these environments deconstructionism-friendly.

« 9 » The idea of adopting the designed-based approach, suggested by Holmes (§6), is interesting. It might have been used in the development of Virtual Classroom if it had been developed as a standalone project. In fact, the development of the environment was governed by specific requirements of the weSPOT project. Additionally, the Virtual Classroom software was not the primary software developed in weSPOT, it was just an add-on facilitating some pilot experiments.

« 10 » Similarly, while I used Mecho, I collected data relevant only to the goals of the corresponding research project. Holmes (§4) is right that many factors may had contributed to the success of Mecho. The instructor in all classes was the same person and he was also the author of all versions of Mecho. So, more or less, the presentation style in all courses was practically the same. Further courses will provide more data and then it will be possible to refine the exact contribution of each factor.

Students' reactions

« 11 » The experience of the students with the software environments was discussed in other papers as part of the dissemination activities of the corresponding projects. Three of these papers are included in the target article (Boytchev et al. 2014; Stefanov Nikolova & Stefanova 2013; Stefanov et al. 2013). Harvey (§11) expressed the importance of social relations in the classroom. The three papers contain further details about what actually happened in the classroom and how students communicated, and I also published some of their conversations.

« 12 » The decision to challenge the students (and their teachers) with the software environment Virtual Classroom without providing any documentation was described as risky in §20. Both Harvey (§10) and Holmes (§5) confirmed it was indeed risky and shared their doubts about its advantages. Of course, I do not claim that such approach will work in every case; I just saw an opportunity in a situation where the lack of documentation might be advantageous. And I utilized this opportunity.

« 13 » In a nutshell, the students experienced no problems with the lack of documentation, they had no problems navigating and managing the virtual microworld. Nowadays, when students try new software or hardware, they spend no time studying its documentation. They just start playing with it right away, using their experiences with other technological tools. When they face problems, they first ask their friends. If the problems are still unresolved, they check the documentation. This answers Holmes' (§5) concerns that constructivist environments require guidance or knowledge. The software was designed in such a way as to match the students' experience with other software (such as games).

Deconstructionism and technology

« 14 » As I talk about technology and how students experience it, I reach another important idea of our article – a change of technology changes how people interact, communicate and explore new things. As a result, it also changes how people (prefer to) learn. Harvey (§8) found it unconvincing that new digital devices, such as tablets, might have significant positive impact on education. In my article I was not concerned with the discrepancy between the current state of technology and the current state of education. I was concerned with the discrepancy between the current speed of change of technology and the current speed of change of education. In this respect, tablets, digital watches and smart glasses are mere artefacts of today's technology. Tomorrow they will be replaced by something else.

« 15 » Technological development is exponential, as noted in the target article (§46) and by Futschek (§7). On the other hand, education is still conservative. It changes, but incrementally. As a result, the gap between technology and education is predestined to become larger unless education makes a significant leap forward. In the past, it was education that empowered people's life, today it is technology (this is a harsh claim, but see the end of §1 of this response).

« 16 » At the end of his comment, Harvey (§11) asked a series of interesting questions that may push the idea of deconstruction further. Additionally, as Holmes (§7) suggested, I shall revisit Derrida's work to look for a stronger relation between his deconstruction and our deconstruction. Although both deconstruction domains are quite different, the conceptual foundation might be quite similar. I am particularly happy with how Futschek (§§4–6) described software development from a deconstruction-construction point of view. I believe that once deconstruction has a name, it will be easier to demystify it and then to use it. Again, being provocative, I see this demystification as recursive activity – to understand deconstruction, one has to deconstruct it.

Received: 22 June 2015
Accepted: 28 June 2015

Combined References

Boytchev P. (2007) Design and implementation of a Logo-based computer graphics course. In: Kalas I. (ed.) Proceedings of 11th European Logo Conference, Bratislava, Slovakia: 21.

Boytchev P. (2013a) Mecho: Educational software for virtual mathematical devices. In: Proceedings of 3rd EDUvision International Conference, Ljubljana, Slovenia: 692–707.

Boytchev P. (2013b) Digital mathematics. In: Proceedings of 42nd Spring Conference of Union of Bulgarian Mathematicians, Borovetz, Bulgaria: 124–133.

Boytchev P., Sendova E. & Kovatcheva E. (2011) Geometry of motion: Educational aspects and challenges. International Journal on Information Technologies and Security 3(1): 27–40.

Boytchev P., Stefanova E., Nikolova N. & Stefanov K. (2014) The virtual classroom: A pilot case in inquiry based learning. In: Zvacek S., Restivo M. T., Uhomoibhi J. & Helfert M. (eds.) Proceedings of 6th International Conference on Computer Supported Education, 1–3 April 2014, Barcelona, Spain. Volume 2. SCITEPRESS Science and Technology Publications, Portugal: 264–269.

Bruner J. S. (1961) The act of discovery. Harvard Educational Review 31: 21–32.

Chemuturi M. (2012) Requirements engineering and management for software development projects. Springer Science & Business Media, New York.

Derrida J. (1989) Memoires for Paul de Man. Columbia University Press, New York.

Design-Based Research Collective (2003) Design-based research: An emerging paradigm for educational inquiry. Educational Researcher 32: 5–8.

Dewey J. (1910) How we think. Heath & Co, Boston.

Kirschner P., Sweller J. & Clark R. E. (2006) Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educational Psychologist 41(2): 75–86.

Lawton G. (2012) Noam Chomsky: Meet the universal man. New Scientist 213(2856): 28–29.

Mayer R. E. (2004) Should there be a three-strikes rule against pure discovery learning? American Psychologist 59(1): 14–19.

Montessori M. (2001) A new world and education. Association Montessori Internationale, Communications 2013/1/2. Available at http://www.montessori-ami.org/articles/article04.pdf. Originally published in 1947.

Protopsaltis A., Seitlinger P., Chaimala F., Firssova O., Hetzner S., Kikis-Papadakis K. & Boytchev P. (2014) Working environment with social and personal open tools for inquiry based learning: pedagogic and diagnostic frameworks. International Journal of Science, Mathematics, and Technology Learning 20: 51–63.

Resnick M. (1990) MultiLogo: A study of children and concurrent programming. Interactive Learning Environments 1(3): 153–170.

Stefanov K. S., Nikolova N. N., Stamenov S., Dimitrova T. & Stefanova E. P. (2013) weSPOT: Inquiry-based science education approach and technologies in action. Annual of Sofia University 101: 123–141.

Stefanov K. S., Nikolova N. N. & Stefanova E. P. (2013) weSPOT: Contemporary approaches in education in natural sciences. In: Proceedings of 42nd Spring Conference of the Union of Bulgarian Mathematicians, Borovetz, Bulgaria: 101–111.

Yahya S., Ahmad E. & Jalil K. (2010) The definition and characteristics of ubiquitous learning: A discussion. International Journal of Education and Development using Information and Communication Technology 6(1): 117–127. Available at http://ijedict.dec.uwi.edu/include/getdoc.php?id=4843