# Deconstructionism – We Do Need More Success Stories

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

**> Abstract** • Deconstructionism in education is a powerful approach, which is still insufficiently researched. Griffin's presentation of the pedagogical aspect of deconstructionism is a success story, which may inspire others. Having experienced some of the issues, I also provide examples from my own practice.

« 1 » This commentary starts with a confession. Five years ago, I presented the concept of deconstructionism in education at the International Constructionism Conference in Vienna (Boytchev 2014). I was scared, as I was about to present a concept seemingly antagonistic to the main topic of the conference. The notion of deconstructionism had only been applied to philosophy and art (Boytchev 2015b: 367). Yet, deconstruction had been used in education for years, but has not been explicitly named and extensively researched. The scare quickly turned into happiness, as deconstructionism gained the attention of other researchers. Thus, this confession is my thank-you to Jean Griffin for providing examples of deconstructionism in the context of pedagogy.

« 2 » Griffin describes several approaches based on deconstructionism. Taking apart machines and appliances (§6), i.e., reverse-engineering, is one of the manifestations of deconstruction, as I have mentioned (Boytchev 2015a: 361). The mechanical dissection (§§8–10) is an accurate application of deconstructionism, although the word "dissection" is misleading. Dissection in medicine is a destructive process, as the topic of study or research is destroyed. Griffin, however, makes it clear that mechanical dissection includes [dis]assembling, re-assembling and developing (§§8-10). The Tod Phod Jod approach is very promising. It openly utilizes the deconstructionist approach. Griffin describes that Tod Phod Jod relies on "fun and intellectually stimulated experience" (§11). This leads me to my first question: What about considering deconstructionism as one of the inborn human behaviours, just like curiosity, creativity, etc? (Q1)

« 3 » Worked examples are another approach and I agree with the conclusion that learning is obstructed by cognitive overload (§13). Problems caused by excessive cognitive load or a cognitive barrier occur in the deconstruction phase (Boytchev 2015a: 362).

« 4 » In the context of programming code, comprehension must precede or occur at the same time as code writing (§16). This conclusion fits with how deconstruction and constructions are coupled. Understanding programming code is a deconstruction process, while writing code is construction. Interactive environments and the codelens component, described by Griffin in §25, provide support for faster and more active code comprehension.

« 5 » Programming topics are fruitful for learning from code and, especially, from intentional errors in the code (§§17f). Griffin provides sufficient references as to why this practice is controversial. My personal experience supports the positive impact of learning by mistakes – both intentional and unintentional mistakes. This approach is embedded in all my courses. One of them, called "Fundamentals of Computer Graphics" (FCG), has errors distributed in many of the lecture notes. This pulls students away from their default "write-only" mode – this is the tendency for students to take notes and to unconditionally soak up whatever information is presented during classes. In this respect I support Griffin's intention to collect more evidence about the potential benefits of learning from errors (§36).

« 6 » Let me recall my early years as a freshman. I enrolled in martial arts classes. The first few months we practiced only how to fall on the tatami in the least painful way, and how to get up immediately. Teaching martial arts has been fine-polished and optimized for many centuries; teaching programming has barely half a century of experience.[1] Can teaching programming learn from teaching martial arts? Do we need to teach students how to recover when their programs break down?

« 7 » I dream of a discipline that not only teaches the correct science, but utilizes the education power of historical misconceptions, failures and dead-ends. For example, in FCG there are historical sections about the catenary curve, studied in 1638 by Galileo, Jungius and Huygens (Kacmarynski 1931: 1). The curve is based on the hyperbolic cosine, but Galileo had mistakenly claimed that it was a parabola (Sonar 2018: 331). Another historical event is the origin of Bézier curves, developed by Paul de Casteljau, but named after Pierre Bézier (Vince 2006: 125) because of academic publishing problems.

« 8 » Griffin's experience in implementing deconstructionism as a pedagogy approach in learning a programming language is another interesting example of why and how deconstructionism is beneficial to education as much as constructionism is. However, I want to bring up a reminder about the importance of the careful designing of educational materials. Programming differs from Math at least in one vital aspect – most programming problems have many solutions – both completely distinct and brutally correct at the same time. This means that a "simple" task like that of locating the exact position of a bug may have multiple correct answers. Worked-out incorrect examples are a powerful arsenal, but they are risky for the teacher and for the students. This is also illustrated in Griffin's model for deconstruction, shown in Figure 4 in the target article and described in §26. All students' tasks within worked-out correct examples ("Explain or Label" and "Discern") could also be accessed via worked-out incorrect examples. The opposite is not correct. The "easier-harder" axis is important as it shows that problems with correct examples are easier than problems with incorrect examples even if they tackle the same student task.

« 9 » The design of suitable problems for the students is a difficult activity. It relies on *empathic deconstruction*. The teacher must foresee the deconstruction processes in students' minds in order to minimize potential failure during the deconstruction at lab-time.

« 10 » The research conducted by Griffin shows positive results. However, I would be more cautious about the conclusions extracted from the survey (§36). Such questions have an inherent risk of subjectivity. I would suggest the author follow up the aca-

1 | See Everett Murdock's lecture notes "History, the history of computers, and the history of computers in education," https://web.csulb.edu/~murdock/histofcs.html

**Figure 1** • The Meiro environment.

demic development of these students and find out whether they have started using Python (of their own accord) in their other classes.

« 11 » Deconstructionism in education is heavily supported by technology. Griffin describes the advantages and disadvantages of using software like Runestone Interactive. I have similar experience with other software and 15 years ago I went along another path by deconstructing the educational software, and then reconstructing it by myself. This path is hard but it provides the freedom to create what you need and the confidence that you can do it when you need to. This is also related to Griffin's plan to introduce appealing graphics and gamification elements (§41). All my courses are "deluged" by my own graphics and interactive 3D models.

« 12 » Since 2018 the FCG course has been enriched by a gamified virtual environment called Meiro (Lekova & Boytchev 2018), which is used to present concepts during lectures and for self-education at home.

« 13 » Figure 1 shows thumbnails of a few hundred 3D models (left) and larger snapshots of the Meiro environment (right). The next version of the virtual environment will have gamified modules for (self-)evaluation and (self-)assessment (Boytchev & Boytcheva 2018). The gamification of Meiro is in the frame of the national project "ICT in Science, Education and Security."

« 14 » I agree with Griffin's conclusions that deconstructionism as a pedagogy is complementary to constructionism, and that the described experiment is a proof of the concept (§43). This commentary started with a confession, and ends with a request for a confession. Knowing a tool (any tool, either hardware, or software, or brainware) is complete, when you know when you *should not* use this tool. So, my second question is as follows: Have there been situations in which deconstructionism had a negative impact on the learning or on the learners? (Q2)

## References

**Boytchev P. (2014)** Deconstructionism in education: A personal wandering towards constructionism. In: Proceedings of the third international constructionism conference. Austrian Computer Society, Vienna: 93–102.

**Boytchev P. (2015a)** Constructionism and deconstructionism. Constructivist Foundations 10(3): 355–363.
▸ https://constructivist.info/10/3/355

**Boytchev P. (2015b)** Author's response: Does understanding deconstruction require its deconstruction? Constructivist Foundations 10(3): 367–369.
▸ https://constructivist.info/10/3/367

**Boytchev P. & Boytcheva S. (2018)** Evaluation and assessment in TEL courses. In: Proceedings of the 44th international conference of applications of mathematics in engineering and economics. AIP Publishing, Melville NY: 020035. https://doi.org/10.1063/1.5082053

**Kacmarynski J. (1931)** The catenary. MS thesis. State University of Iowa. https://doi.org/10.17077/etd.o2vdr7xz

**Lekova M. & Boytchev P. (2018)** Virtual learning environment for computer graphics university course. In: Proceedings of the 12th international technology, education and development conference. IATED Academy, Valencia: 3301–3309.

**Sonar T. (2018)** The history of the priority dispute between Newton and Leibniz: Mathematics in history and culture. Birkhäuser, Basel.

**Vince J. (2006)** Mathematics for computer graphics. Second edition. Springer, London.

**Pavel Boytchev** is an associate professor and researcher at the Faculty of Mathematics and Informatics, Sofia University. His research interests are in the areas of developing university-level courses, educational software, computer graphics and multimedia. He has created numerous educational applications based on his own educational programming languages. He is an author of a dozen courses, hundreds of computer-generated video clips and thousands of computer demo programs in his areas of interest.

Received: 27 May 2019
Accepted: 31 May 2019