Computers in Human Behavior 61 (2016) 427–442

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Full length article

A case study of undergraduate engineering students' computational literacy and self-beliefs about computing in the context of authentic practices

Alejandra J. Magana [a, *], Michael L. Falk [b], Camilo Vieira [a], Michael J. Reese Jr. [b]

[a] Purdue University, USA
[b] Johns Hopkins University, USA

ARTICLE INFO

Article history:
Received 13 January 2016
Received in revised form
4 March 2016
Accepted 8 March 2016
Available online 25 March 2016

Keywords:
Computational literacy
Self-beliefs
Modeling and simulation practices
Anchored instruction

ABSTRACT

Engineering students, as compared to computing-related majors, are not traditionally introduced to computing in the context of authentic learning experiences, i.e., real-world applications within their discipline. This paper identifies the impact of computation delivered by authentic learning experiences in the form of anchored instruction on students' self-beliefs and their capacity to leverage computation to acquire disciplinary concepts in subsequent computationally-based engineering coursework. This case study included 130 students with different programing preparation (authentic or traditional), who were exposed to computational learning modules. Control-Value Theory of Achievement Emotions is the conceptual framework that guided the evaluation of this investigation. Measures included student self-beliefs such as control and value appraisals, and their relationship with academic performance. Results suggest that programming preparation presented in an authentic engineering context provides an important foundation that goes beyond increasing students' control self-beliefs. This preparation seems to effectively enable students to leverage computational practices for the purpose of acquiring disciplinary concepts. Implications for teaching relate to the integration of computation sooner, more often and within a disciplinary context in the undergraduate engineering curriculum. Implications for learning relate to fostering engineering computational literacy guided by anchored instruction to support disciplinary problem solving.

1. Introduction

An important component for producing successful outcomes in engineering design and innovation is analytical problem solving ability, which remains central to engineering education (Litzinger et al., 2010). These analytical aspects supplement known best practices that include iterating through all the steps of the problem solving process, developing multiple solutions to the problem, and gathering sufficient information (Atman, Chimka, Bursic, & Nachtmann, 1999). Modern engineering workplaces now use modeling and simulation practices, coupled with computational tools, to aid the analysis and design of systems (Emmott & Rison, 2008; McKenna & Carberry, 2012). Thus, these practices and tools have been widely acknowledged by the engineering community as important.

Early steps in the problem solving process involve generating the model of the system being studied and identifying a set of equations describing the system. It also involves using the model to propose a solution and then evaluating and validating the solution (Litzinger et al., 2010). Facility with domain-specific software and computational tools has become an essential form of literacy for participating in the engineering problem solving process due to the increased complexity of engineered systems. We argue that engineering computational literacy consists of modeling and simulation processes, along with the underlying computational methods and techniques to actualize them. Furthermore, domain-specific software and computational tools for analysis and design in engineering have reached a crucial level of development for integration into the learning cycle. Used in the context of authentic learning experiences, these tools can now support the integration of multiple cognitive processes and skills such as divergent and

* Corresponding author. Knoy Hall Building, Room 231, 401 N. Grant Street, West Lafayette, IN 47907-1421, USA.
E-mail address: admagana@purdue.edu (A.J. Magana).

http://dx.doi.org/10.1016/j.chb.2016.03.025
0747-5632/

convergent thinking (Dym, Agogino, Eris, Frey, & Leifer, 2005) within modeling and simulation processes relevant to engineering design and can accelerate the acquisition of core engineering concepts. These skills have been adopted across many disciplines as analytic tools that support the analysis of complex phenomena and as predictive tools that can anticipate the suitability of new designs.

Recognizing the importance of these skills, engineering education policymakers and the engineering education research community have recommended their incorporation into the curriculum. For instance, the Transforming Undergraduate Education in Engineering report ([ASEE], 2013) recently identified that industry professionals value programming skills and the ability to use computational tools to support problem solving. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice is a stated ABET (2013) student outcome. The Washington Accord ([IEA], 2011) stipulates program outcomes including (a) the application of knowledge of mathematics, science, engineering fundamentals to the conceptualization of engineering models, along with (b) the creation, selection and application of appropriate techniques, resources, and modern engineering tools, including prediction and modeling, to complex engineering activities, with an understanding of the limitations.

It has been widely acknowledged that engineering instructors and instructional designers are not keeping pace with the need for graduates with this complex skill set ([NRC], 2011; [WTEC], 2009; Emmott & Rison, 2008; Guzdial, 2011). For example, students who decided to pursue a computational concentration in their graduate degrees expressed frustration and concerns about not being prepared to solve highly interdisciplinary and highly computationally complex problems (Magana & Mathur, 2012). Students repeatedly reported a lack of common basic computational knowledge and insufficient exposure to computational engineering (i.e., computation to solve problems in their discipline) at the undergraduate level, receiving such training only late in their academic careers, i.e. at the master's or doctoral level (Magana & Mathur, 2012). These research findings imply that acquiring computational literacy is a process that must be acquired over the course of a student's academic and professional career.

Experiences deploying programming, computational tools and software in the context of their discipline through repeated exposure may be crucial for students' acquisition of these highly desired skills. Also, students' self-perception of their abilities and interests in computation, which are arguably critical for their decisions to pursue careers that depend on such skills, may be quite sensitive to the nature of their exposure to programming, computation and software. For example, studies focusing on first-year students, have identified that computing abilities, problem-solving abilities, and understanding of disciplinary concepts, among others, are sources of confidence of engineering students' self-efficacy; where computing abilities are amongst the most influential (Hutchison-Green, Follman, & Bodner, 2008; Hutchison, Follman, Sumpter, & Bodner, 2006).

In this study, we specifically seek to explore the effect of the foundational computational preparation at the undergraduate level on students' self-beliefs and their ability to deploy computation to acquire conceptual knowledge in their field. In particular, we wish to understand whether "authentic" computational learning, i.e. learning situated within the context of their engineering discipline, early in students' undergraduate career (e.g., freshmen/sophomore) has an influence on their beliefs and performance in subsequently programming, applying computational tools, and using application software. It is important to note that this modality is not the norm currently, as most students are initially introduced to computation as a distinct discipline and only late in their academic careers, if ever, do they have the opportunity to deploy computation in an engineering context.

The context of this study is a Materials Science and Engineering (MSE) department at Johns Hopkins University that wanted to address this educational deficiency by preparing its students for the increasing role computation plays in the discipline. It implemented a curricular innovation across the core courses of the undergraduate program to introduce computer programming, modeling and simulation practices, and the application of engineering tools. The curricular innovation consists of a new discipline-based gateway computing course entitled "Computation and Programming for Materials Scientists and Engineers" (CPMSE), coupled with the integration of computational learning modules in the major's six core courses. The CPMSE course's primary learning goal is for students to apply algorithmic thinking and computer programming toward the solution of engineering and scientific problems relevant to MSE (Magana, Falk, & Reese, 2013). The modules integrated into the subsequent core courses are designed to reinforce computational materials science and engineering (CMSE) skills and to facilitate the acquisition of foundational MSE concepts.

This study describes undergraduate materials science and engineering students' shifting self-beliefs and learning gains that resulted from the curricular innovation. Specifically, we investigated students' self-perceived abilities at performing various modeling, simulation and computation tasks, their perceived value of these skills in their academic and future professional careers, as well as the effectiveness of this approach to improve the learning of disciplinary concepts. Specific research questions for this study are:

1. Are there any differences in self-beliefs among students who have different prior programming experiences?
2. Are there any differences in disciplinary learning among students who have different prior programming experiences?
3. How do students' prior programming experiences relate to their self-beliefs and their disciplinary learning?
4. How do students' self-beliefs relate to each other and their disciplinary learning gains after exposure to authentic computational learning experiences?

2. Student self-beliefs and academic achievement

Educational researchers and psychologists, among others, have extensively investigated student self-beliefs as related to academic achievement. Self-belief constructs (i.e., self-efficacy, self-concept and self-esteem) describe a person's opinions of his or her own attributes and abilities as an individual (Bryne, 1996; Valentine, DuBois, & Cooper, 2004). A meta-analysis of longitudinal studies aimed at finding the effect of self-beliefs, identified a small, favorable influence of student self-beliefs on academic achievement (Valentine et al., 2004). Furthermore, stronger effects were identified when researchers evaluated self-beliefs associated to a specific domain or discipline (Valentine et al., 2004).

In engineering education contexts it has been suggested that specifically, self-efficacy strongly influences choices engineering students may make (Hutchison-Green et al., 2008). These influences can be positive or negative based on their perceptions of their abilities as related to a particular task (Hutchison et al., 2006). Similarly, research suggests that students' self-efficacy, is positively related to indicators of cognitive engagement (Greene & Miller, 1996). That is, students who feel more confident in their ability to perform a given task are more likely to engage their repertoire of learning strategies and persist at using those. Exploratory studies conducted at the freshmen level have identified that students' self-efficacy and anticipated future success decreased over the first year for both men and women (Jones, Paretti, Hein, & Knott, 2010).