Freedom of Choice as Motivational Factor for Active Learning

Atanas Radenski
Chapman University
Orange, CA 92869, USA
radenski@chapman.edu

Freedom is nothing else but a chance to be better ~ Albert Camus

ABSTRACT
Freedom to choose what, when, and how to contribute in a learning process can motivate students to actively engage and achieve more in their studies. However, freedom of choice complicates course management and may deter instructors from allowing such freedom. Our approach is to utilize existing functionality of course management systems such as Moodle to automatically facilitate and coordinate free student choices and provide much needed relief for instructors at the same time. Using Moodle we have developed novel digital study packs that blend freedom of choice with guidance and control. Our survey shows that *assisted freedom of choice* is ranked highest in 51% of student responses – in contrast to *unlimited choice* at 28% or *no choice at all* at 21%. Experience reported in this paper may be beneficial for instructors who would like to expand their courses with new motivational learning techniques.

Categories and Subject Descriptors
K.3.2 **[Computers & Education]**: Computer and Information Science Education - *computer science education, curriculum*; Computer Uses in Education - *distance learning*

General Terms
Human Factors, Languages

Keywords
Active learning, programming languages, CS1/2, study pack, Moodle, compiler construction, Java, Python, labs, projects

1. RATIONALE
The dot-com bubble burst in years 2000-2002 was followed by a decline of IT employment and by a corresponding decline of CS/CE enrollment in the US and Canada.

The *IT employment* numbers peaked in 2002, declined in 2003, then increased steadily since 2003 to grow 6.9% higher in 2007 than in 2001 [15]. Conversely, the number of newly declared *CS/CE undergraduate majors* peaked in 2000 and then declined 46% between 2002 and 2005 alone, according to the authoritative Taulbee Surveys[1] [7]. A slight enrollment increase in 2006 was followed by another worrisome decrease in 2007. These employment and enrollment data are shown in Figure 1.

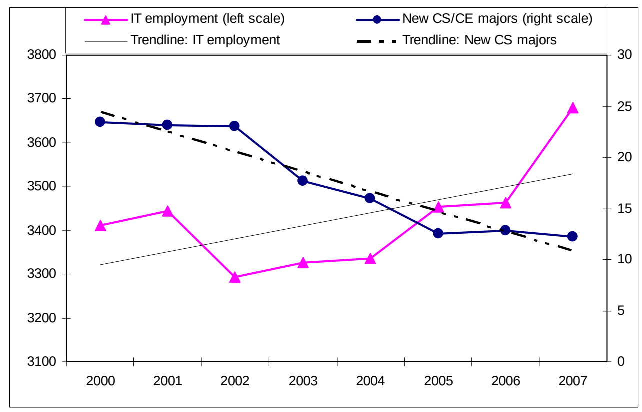

Figure 1. IT employment [15] and newly declared undergraduate CS/CE majors [7], in thousands.

The steady IT employment growth has not stimulated reciprocal growth in CS enrollment yet, as some have expected. It is more important than ever to seek ways beyond job market factors to improve CS enrollment. The need to *stimulate and increase interest* in CS has been addressed from various perspectives, some of which are outlined in the following list.

Innovative pedagogy. Educators develop pedagogical approaches intended to boost student interest, motivation, and satisfaction, such as active learning [4, 9, 12, 17, and 18], team-based learning [8, 11, and 12], and cooperative learning [25].

Innovative course development. Educators develop courses that involve game development, virtual reality, multimedia, robotics, and the web [1, 3, and 8]; inter-disciplinary courses [10]; introductory courses that make programming easier to master [18]; introductory courses or course modules that do not involve programming at all [6]; course clusters that provide multiple-entry points in computer-related majors [16] - and many others.

Use of emerging information technologies. Educators free students to learn independently of time and location by means of course management systems (CMS) such as Moodle [18], by means of Web-based tools [9], and/or by means of mobile computing and communication devices, such as mobile phones and MP3 players [12]. Educators experiment with promising pedagogical uses of

"© ACM, 2009. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Proc. ITiCSE '09, ACM, New York, NY, 21-25. DOI= http://doi.acm.org/10.1145/1562877.1562891.

[1] Survey data are on undergraduate enrollment from Ph.D.-granting departments of computer science (CS) and computer engineering (CE) in the US and Canada.

tablets [4 and 20] to support active learning and harness the growing popularity of social networks [3] to engage students in the learning process.

Addressing the gender gap. Educators investigate factors for the declining interest of women in IT careers [13] and specifically in CS [23]. CS educators experiment with practical methods to boost women's interest in CS [22].

Advertising and recruitment. Administrators and faculty offer practical methods to advertise and recruit among future students and students who are already on campus [24]. Scholars develop theoretical models to explain and predict student motivation and develop corresponding recruitment strategies [2].

The above approaches are usually combined rather than used in isolation. For example, tablet PCs are employed in active learning [4 and 20] and gender issues are addressed by multimedia-rich pedagogy [19].

Our general goal is to *stimulate student interest, motivation, and satisfaction* through (1) lab-based and project-based course development to promote (2) active learning pedagogy supported by (3) emerging information technologies such as the Moodle CMS. Similarly to others [4 and 9], we aim to achieve this goal through a combination of emerging information technologies, pedagogy, and course development. Our specific choices are justified as follows.

- In the realm of *emerging information technologies*, we focus on Moodle because it is an open source CMS that is increasingly popular with educational establishments - currently (January 2009) there are over 47,000 registered Moodle sites [14] in 199 countries with nearly 2.5 million courses with about 25 million students.
- In the realm of *innovative pedagogy*, we focus on active learning because of its increasing recognition as a method to boost student involvement and interest.
- In the realm of *innovative course development*, we focus (1) on a lab-based approach because it has been recognized to be beneficial at the introductory CS1/2 level and (2) on a project-based approach because it is well known to be productive at the advanced undergraduate level.

Freedom to choose what, when, and how to contribute in a learning process can motivate students to actively engage and achieve more in their studies [5].

Note however that freedom of choice complicates course management and may deter instructors from allowing such freedom. For example, an instructor may give the same assignment to all students simply because guidance and evaluation of a variety of custom, individually selected assignments would require more time and effort. Our approach is to utilize existing functionality of CMS environments such as Moodle to automatically facilitate and coordinate free student choices and provide much needed relief for instructors at the same time. Using Moodle we have developed novel digital study packs that blend freedom of choice with guidance and control.

The goal of this paper is to present rationale (in part 1), describe an implementation (in part 2) and offer an evaluation (in part 3) for *freedom of student choice* as a motivational factor in active learning.

2. IMPLEMENTATION

2.1 Digital Study Pack Overview

We have implemented four new courses that provide students with a variety of *free choices* in order to stimulate student interest, motivation, and satisfaction. These four courses were originally designed and taught at Chapman University in California, USA. Two of these are upper-level project-based courses on (1) Programming Languages and (2) Compiler Construction, and the other two are introductory-level lab-based courses - (3) CS1 with Python and (4) CS2 with Java.

In support of these courses, we have designed *online study packs* which are comprehensive collections of digital resources (such as e-texts, tutorials, and slides) and activities (such as homework, self-tests, databases, forums, and messaging). The home page of a study pack contains a list of *topics*, together with links to resources and activities for of each topic (Fig. 1).

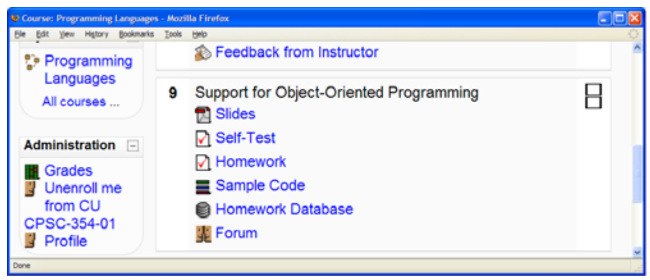

Figure 1. Topic *Support for Object-Oriented Programming* from the *Programming Languages* study pack home page

A study pack template can be replicated and customized to support various course sections at the same school or at different schools. All study pack templates and their replicas are implemented within a Moodle installation at http://studypack.com. Each study pack instance is pre-programmed by the corresponding instructor with all deadlines and is made available to students in its entirety at the beginning of their course of study.

Courses that are supported by comprehensive online study packs are not necessarily online courses (although they can be). In fact, **our digital study packs have been exclusively used in scheduled on-site courses** at Chapman University and at seven other schools in the US and abroad. Study-pack based courses involve scheduled lecture/lab meetings and utilize paper or digital textbooks. For example, the Programming Languages study pack is based on a well-established paper textbook [21]. Instruction is based on lectures, a semester-long project, self-study (reading slides and text-book chapters), online self-tests, online homework, discussion forums, and in-class exams. The online study pack is used like a virtual workbench where students find digital resources and perform required and optional tasks.

2.2 Freedom of Choice with Study Packs

An online digital study pack is an all-in-one e-learning solution that is constantly available to students and instructors, independently of time and location. A digital study pack frees students with busy schedules to actively engage in learning activities not when they are told by the instructor to do so, but when they have the time and desire to do so. For example, some students choose to work at night while others chose to work during the day. In addition, students are free to choose exactly what to do in a particular work session. For example, one student may choose to do a late-night self-test while another student may prefer to do homework at the same time.

In this paper we focus preferentially on our Programming Languages project-based study pack (1) because of space constraints, (2) because this study pack is the newest one and represents the current state of our work, and (3) because principal features of our lab-based CS1/2 study packs have already been published elsewhere [18]. We also offer brief overviews of the other three study packs.

The **Programming Languages study pack** evolves around fundamental topics such as expressions, control structures, abstract data types, OOP, concurrency, exceptions, and others. Along the course of study, each student explores such topics using their own individually chosen programming language. The student carries out experiments and develops a sequence of homework programs in their chosen language, one program for each topic. In addition to homework, the student uses the same language to develop a major project program, write a project paper, and prepare a project presentation.

All of these activities involve significant *free choices*: each student is free to select his or her own preferred programming language and what kind of programs to develop for homework assignments and for his or her own semester project. All these free choices are facilitated and coordinated by the Programming Languages study pack through carefully programmed activities and resources.

The choice of a programming language, for example, is shaped as a special online homework which provides extensive guidance and demands for particular activities. Students follow posted guidance to acquaint themselves with a list of recommended languages; explore additional languages; freely choose a preferred language; identify and freely select their preferred documentation; pick, retrieve and install their preferred programming environment; identify, design, and implement preferred a sample program; and run it. At the end of the homework, students upload their individual results in a Moodle database where they can be observed by the instructor and by their peers. As students complete and submit the homework form, they automatically receive provisionary credit. The homework and/or the provisional credit are subject to corrections by the instructor (Fig. 2).

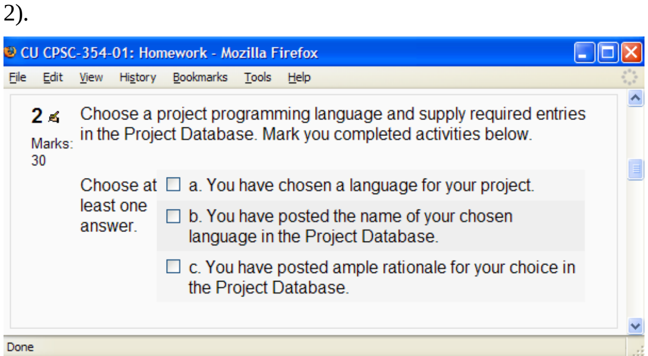

Figure 2. Programming language selection homework (partial)

In addition to free choices along the semester-long project, students make individual choices in the study of individual topics. For example, the homework on *concurrency* calls for the study of the principal concurrency features of unit-level concurrency, competition synchronization, and cooperation synchronization. All students are asked to answer the same concurrency-related questions but in the context of their individually chosen programming languages (Fig. 3). In the process, each student is free to design his or her own preferred concurrency examples and a complete concurrent sample program in his or her chosen language. Results are posted in the current topic forum and program sources and screenshots are posted in the current topic database. All results can be observed by the instructor and by all students. In addition, results are reported in informal oral presentations given by students in class, prior to new topic lectures. Because student submissions are open to everyone from the class, students browse and study them, actively provide comments and feedback in forums, and learn from each other.

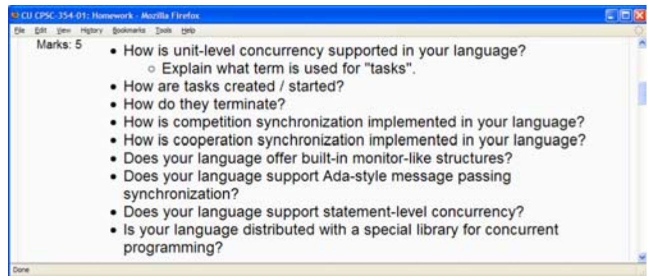

Figure 3. Questions from the *concurrency* homework (partial)

A project-based **Compiler Construction study pack** has the same architecture as the Programming Languages pack. This pack focuses on the topics of scanning, parsing, abstract syntax trees, code generation, and virtual machines. The course evolves around the study of a sample educational language and compiler. Each student is supposed to extend the educational language with a new interesting feature, such as a new statement or new type. Students gradually implement their preferred language extensions over the course of study, topic after topic. During the study of parsing, for example, students implement and report parsing routines for their chosen language extensions. The entire learning process in general - and free choices in particular - are guided by the study pack through preprogrammed activities and resources.

Two lab-based study packs, the **CS1 with Python study pack** and the **CS2 with Java study pack**, employ self-guided labs, e-texts, tutorials, quizzes, and forums to introduce principal topics such as control, functions, objects, classes, I/O, exceptions, GUI, and graphics. Numerous self-guided labs promote active learning by leading students in suitably designed programming experiments and development. For example, the CS1 with Python pack contains a self-guided programming lab which guides students on how to use stepwise refinement in the development of a GUI.

Self-guided labs support three programming modes that suit students with different backgrounds. First, inexperienced students may follow complete and detailed prescriptions of what to do and how to do it. Second, experienced and motivated students may acquaint themselves with the lab specification and then develop the required software by themselves. A third category of students may try to find a solution independently while peeking into detailed instructions when help is needed. All students are free to choose how much guidance to follow and how much to challenge themselves and seek solutions on their own.

3. EVALUATION

In the 2008 spring and fall semesters, we administered a survey of student and instructor perceptions of the freedom of choice as motivational factor in various activities.

The survey was offered (1) to students from four lab-based CS courses at Chapman University, California; Columbus State University, Georgia; and Berry College, Georgia, and (2) to students from project-based programming language courses at Chapman University, California and Sofia University, Bulgaria. In addition, the survey was offered to instructors from four universities, two colleges, and four high schools in the USA.

All surveyed students were *study pack learners* during or immediately before the survey. All surveyed instructors are current or past *study pack adopters*. Therefore, all survey results should be interpreted in the context of online study pack usage, as outlined in Section 2 of this paper. In the survey, we received 32 responses form a group of 80 learners and 8 responses from a group of 10 adopters (instructors).

Answers to the first survey question demonstrate that instructor-assisted freedom of choice in semester projects is perceived as a great motivational factor by all categories of study pack users: introductory level students, advanced students, and instructors (Table 1). At the same time, projects that are mandated by the instructor are believed to be least motivational and engaging.

Table 1. Semester projects and freedom of choice

Students get motivated and engaged in **projects** when [2]:	***Students:*** *Intro level*	***Students:*** *Advanced*	***Instructors***
- every student from the class must work on the same project as specified by the instructor	<u>2.0</u> [3]	2.5	2.5
- every student is obliged to work on a specific, individual project selected by the instructor	2.3	<u>2.3</u>	<u>2.2</u>
- students must select individual projects from a list of permitted choices provided by the instructor [4]	2.3	2.7	2.9
- students are free to select their individual projects by following some guidance from the instructor [5]	**4.5** [3]	**4.4**	**4.0**
- students are free to select their individual projects as they wish, without any guidance or limitations	3.3	4.0	2.8

Surveyed students provided positive free-form feedback, like this:

- *"Firstly, I realized how much fun I have had with this project. Being able to take a language and explore it, study it, and learn it is both exhilarating and challenging. The freedom to choose the final project was an immense factor in my devoting so much time and effort into it. Not only was I learning, but I felt a sense of pride and ownership, as I was free to work on my topic of interest" – Joe Smith*

Table 2. Homework and freedom of choice

Students get motivated and engaged in **homework** when [2]:	***Students:*** *Intro level*	***Students:*** *Advanced*	***Instructors***
- every student from the class must work on the same homework as specified by the instructor	2.7	<u>2.8</u>	**3.8**
- every student is obliged to work on specific, individual homework activities selected by the instructor	<u>2.5</u>	3.1	3.6
- students must select their homework activities from a list provided by the instructor	3.2	3.2	3.5
- students are free to select their individual homework by following some guidance from the instructor [6]	**4.4**	**3.9**	3.4
- students are free to select their homework activities as they wish, without any guidance or limitations [7]	3.6	3.7	<u>2.0</u>

Table 3. Labs (programming assignments) and freedom [2]

Students get motivated and engaged in **labs (programming assignments)** when [2]:	***Students:*** *Intro level*	***Students:*** *Advanced*	***Instructors***
- students must do labs in class and according to a schedule provided by the instructor	<u>2.1</u>	<u>2.8</u>	**3.8**
- students are free to select when and where to do labs, but in compliance with deadlines set by the instructor	**4.5**	**3.9**	**3.8**
- students are free to select when and where to do labs as they wish[8], without deadlines	3.0	3.8	<u>2.5</u>

While students and instructors concur regarding projects (Table 1), the two groups diverge on homework. Answers to the second survey question reveal that instructor-assisted freedom in homework is perceived, on average, as the most important motivational factor by

[2] numbers characterize *importance* for motivation on a 1 to 5 scale: 1 means *least important* and 5 means *most important*

[3] minimal values are <u>underlined</u>; maximal values are in **bold**

[4] without any choices outside of the list

[5] such as a sample list of possible projects

[6] such as a list of the most important topics from the chapter

[7] as long as homework activities are related to the current chapter

[8] before the end of the semester

study pack students (Table 2). In a striking contrast to students, instructors seem to believe that a single mandatory homework can be very motivational for students – a view that is clearly rejected by students themselves (Table 2).

Answers to the third survey question reveal that the freedom to select when and where to do labs and programming assignments is recognized as a serious motivational factor by all categories of study pack users (Table 3). Students would like to be free to select when and where to do labs, but in compliance with deadlines set by the instructor. Students seem to dislike traditional scheduled labs that are limited to class meetings, while instructors seem to oppose unstructured labs and programming assignments without firm deadlines.

4. CONCLUSIONS

An instructor who teaches an onsite or online course by means of an online study pack directs the learning process largely behind the scenes – initially by scheduling and programming the study pack with deadlines before the course and then by facilitating students during the course. Throughout the course, the study pack offers students substantial freedom of what to do and when to do it; such freedom motivates students to actively engage in the learning process and do more.

Instructors who adopt **ready-to-use online study packs** for their courses can utilize preprogrammed lab and project guidance, minimizing the need to impart it to each student individually. Study pack guidance takes many different forms, such as automatically enabled deadlines, detailed self-guided labs, and project specifications. Digital study packs are designed to automatically coordinate and facilitate active learning processes and in doing so, they free the instructor from tedious and time consuming activities.

Our survey shows that *assisted freedom of choice* is ranked highest in 51% of student responses – in contrast to *unlimited choice* at 28% or *no choice at all* at 21%.

Experience reported in this paper may be beneficial for instructors who would like to expand their courses with additional active learning techniques.

5. REFERENCES

[1] Adams, J., J. Holtrop. Building an economical VR system for CS education. *ITiCSE '08*, Madrid, Spain, 2008, 148-152.

[2] Akbulut, A., C. Looney. Inspiring students to pursue computing degrees. *Communications of the ACM*, Vol. 50, No 10, 2007, 67-71.

[3] Alt., C., O. Astrachan, J. Forbes, R. Lucic, S. Rodger. Social networks generate interest in computer science. *SIGCSE'06*, Houston, Texas, USA, 438 – 442.

[4] Anderson, Richard, Ruth Anderson, K. Davis, N. Linnel, C. Prince, V. Razmov. Supporting active-learning and example-based instruction with classroom technology. *SIGCSE'07*, Covington, Kentucky, USA, 2007, 69-73.

[5] Becker, K. How much choice is too much? *Inroads*, Vol. 38, No 4, 2006, 78 - 82.

[6] Cortina, T. An introduction to computer science for non-majors using principles of computation. *SIGCSE 07*, Covington, Kentucky, USA, 2007, 218-222.

[7] CRA Taulbee survey. Retrieved in December 2008 from http://www.cra.org/statistics/.

[8] Goulding, T., R. DiTrolio. Complex game development by freshman computer science majors. *Inroads*, Vol. 39, No 4, 2007, 92-99.

[9] Hamada, M. Web-based tools for active learning in information theory. *SIGCSE'07*, Covington, Kentucky, USA, 2007, 60-64.

[10] Herath, S., A. Herath, R. Kachur, S. Herath. An interdisciplinary accounting forensics course to improve computer science enrollments. *Journal of Computing Sciences in Colleges*, Vol. 23 , No 6 , 2008 ,74-81.

[11] Huss-Lederman, S., D. Chinn, J. Skrentny. Serious fun: peer-led team learning in CS. *SIGCSE'08*, Portland, OR, USA, 2008, 330-331.

[12] Lindquist, D., T. Denning, M. Kelly, R. Malani, W. Griswold, B. Simon. Exploring the potential for mobile phones for active learning in the classroom. *SIGCSE'07*, Covington, Kentucky, USA, 2007, 384-388.

[13] McKinney, V., D. Wilson, N. Brooks, A. O'Leary-Kelly, B. Hardgrave. Women and men in the IT profession. *Communications of the ACM*, Vol. 51, No. 2, 2008, 81-84.

[14] Moodle statistics. Retrieved in January 2009 from http://moodle.org/stats/.

[15] Panko, R. IT Emplyment Prospects: beyond the dot-com bubble. *European Journal of Information Systems*, 2008 (to appear).

[16] Pearce, J., M. Nakazawa M. The funnel that grew our CIS najor in the CS desert. *SIGCSE'08*, Portland, OR, USA, 2008, 503-507.

[17] Prince, M. Does active learning work? A review of the research. *Journal of Engineering Education*, Vol. 93, No 3, 2004, 223-231.

[18] Radenski, A. Digital support for abductive learning in introductory computing courses. *SIGCSE 07*, Covington, Kentucky, USA, 2007, 14-18.

[19] Rich, L., H. Perry, M. Guzdial. A CS1 course designed to address interests of women. *SIGCSE'04*, Norfolk, Virginia, USA, 190 – 194.

[20] Roschelle, J. D. Tatar, S. R. Chaudhury, Y. Dimitriadis, C. Patton, C. DiGiano. Ink, improvisation, and interactive engagement: learning with tablets. *IEEE Computer*, Vol. 40, No 9, 2007, 42-48.

[21] Sebesta, R. *Concepts of Programming Languages.* Pearson Education, Inc., 2008.

[22] Townsend, G., S. Menzel, K. Siek. Leveling the CS1 playing field. *SIGCSE'07*, Covington, Kentucky, USA, 2007, 331-335.

[23] Voyles, M., S. Haller, T. Fossum. Teacher responses to student gender differences. *ITiCSE '07*, Dundee, Scotland, 2007, 226-230.

[24] Walker, H. Advertising and recruiting. *Inroads*, Vol. 40, No 2, 2008, 16-17.

[25] Bartelsen, Jan. Entwurf einer kooperativen Lehrveranstaltung in moodle für die Grundausbildung in der Programmierung. Master-Thesis, University of Hagen, Germany, 2008.