SOFTWARE SKILL PORTABILITY: A CASE STUDY

A. K. Aggarwal, University of Baltimore, Baltimore, MD, USA; 410 837-5275

ABSTRACT

As internet expands more and more students are opting for on line courses. On-line classes create challenges for professors who teach analytical, programming and/or software courses. On-line courses are not meant for students that require additional or extra attention. This problem magnifies in courses that are either software based or use software extensively. Many students have working knowledge of some of the popular software like EXCEL, WORD etc. The problem arises when a different software is used in the class. This raises a basic question: Can students familiar with one software learn a similar software on their own or in short time? Learning here implies learning the software syntax and its application. This paper discusses software skill portability in an online database course. Based on our experience we found that some students proficient in a software do adapt to similar software easily.

INTRODUCTION

As economy falters, resources are becoming fewer and classes are becoming larger. This is putting additional burden on instructors who must maintain education quality with limited resources. This problem magnifies when courses involve analytical and/or programming skills. These problems are further magnified if classes are online. Students are taking many courses online without realizing the extra burden that places on them. Many students do not understand the rigor of online courses and take them as "add-on" courses. This makes teaching software oriented courses challenging for instructors. Not only students have to learn the syntax of software, they have to apply them in business setting. On the bright side, many students already have software skills which may save time if students are able to transfer their skills in similar software. This would free instructor to concentrate on the concepts as opposed to the syntax of the software. This paper studies software portability issue in a data base course.

The next section briefly reviews portability and the following section describes the experiment and the final section discusses the results.

SOFTWARE SKILL PORTABILITY

Portability has been defined extensively in terms of software portability, i.e., a software is portable if can run on different platforms with little or no modifications. According to Garen (2007), "..Portability does not imply that the application will transfer simply to the new platform, but nor does it mean a major effort is required. Portability is simply the ability to use the program on another system with reasonable cost and effort..". Portability has also been defined in terms of employees transferring skills to other employees. Portability has also been defined in terms of hardware and software. Universal remote control (same control can be used with many different devices), laptops (portable with respect to time and space) are some example of portable devices. However, there is very little literature on software skill portability, the topic of this paper. Only recently, literature has started to emerge in this area. Tom Stern (2008), emphasizes the need for skill portability, According to him "..In today's culture of reduced job security with a greater emphasis on skill-set development and portability, people aren't just *taking* jobs; they feel the necessity to use them. ..".

Portability is becoming important at many levels. In general portability could imply transferring software skills from one course to another course or from one software to another software or from one version to an updated version of the same software. (See table 1)

Table 1: Software Skill Portability

Portability level	Example
Software Skills	
• One software to another (similar) software	ACCESS to ORACLE
• One version to another	MS-office version 2003 to 2007
Pedagogical Skill	
• Applying to different area	Spreadsheet skill learned in a course can be applied to MIS, FINANCE etc
• Apply same skills to different versions	Programming skills applied to different programming languages, VBASIC, C++ etc

Software skill portability is defined as skill that learned once can be used repeatedly in other software. Skill portability can be defined in many different ways. We have defined it at two levels:

- Software—these refers to software skills that are transferred from software to software
- Pedagogical skill—these refer to pedagogical skills transferred from one situation (course) to another

In this paper we will study software skill transferability at the software. The next section discusses the experiment.

THE EXPERIMENT

The present study was conducted at an urban public university in the Mid-Atlantic area. The university is an upper-division university and has a non-traditional, commuter student population. University offers BS in MIS, BBAs with MIS specialization and MBA with MIS specialization. The database course was selected to study software skill portability. Database course has first introductory MIS course as a pre requisite but in reality many students also finish at least one programming language course before they take this course. The purpose, here, is to learn if one software related competency can be used in other software. Many students come with ACCESS experience from community college or from work, however, the course uses ORACLE software. This poses many challenges. Can student learn ORACLE software skills on their own since they already know ACCESS? Can student apply ACCESS pedagogical skills in ORACLE? If students can learn software on their own then instructor can concentrate on other important topics. The following hypotheses were developed:

H1: Students who had prior knowledge of ACCESS software package will do better in the course.

Selection of learning objectives

Though there are many learning objectives (LO) for the database course, we will concentrate on the following:

Upon completing this course students will be able to:

1. Implement database systems into database software.

In this LO, students should be aware of syntax and programming of the software used. This relates to "Knowledge" and "Application" category of Bloom's taxonomy (Bloom, 1956). Our assessment of student learning will focus on database syntax knowledge and application portability.

Selection of assessment approaches

Typically, there are several aspects of software competency – a) software accessibility b)

understanding the syntax, c) correcting "bugs" and d) developing applications

Software Accessibility
Implies ease with which students can access the software remotely. Ftp related instructions are provided.

Understanding the syntax
Software have their own syntax and ORACLE is no exception. ORACLE has its own syntax that is used in creating tables, writing queries in SQL etc. It relates to student's understanding of syntax.

Correcting "bugs"
Removing errors. Making sure that SQL statements are correct. This relates to understanding of "logic" of SQL

Developing complete programs
Developing complete solution to problems.

We will use the following measures for software portability:

- Questionnaires: at the beginning and end of the course to measure student's perception of their database software competency. A scale of 1-10 was used for this purpose.
- ORACLE assignments to measure software skill portability.

Assessment Phase

Thirty seven students in database class were surveyed at the beginning of the semester. The questionnaire was divided in several sections:

1. Demographic information
2. Familiarity with e-learning environment
3. Self-efficacy skills in the following areas
    i. Internet browsing
    ii. Spreadsheet analysis
    iii. File transfer protocols (FTP)
    iv. On-line accounting research databases
    v. Locating company financial statements on-line
    vi. Database competency and

For this experiment, we will only look at database competency to study skill portability.

Collect data

Students’ Backgrounds

Students were given 2% of their grade for completing the questionnaire at the beginning and ending of the semester. Table 2 summarizes student’s perception of database proficiency.

Table 2 (a) Database Competency

A_db_create

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	5.4	5.7	5.7
	2	3	8.1	8.6	14.3
	3	5	13.5	14.3	28.6
	4	3	8.1	8.6	37.1
	4.5	1	2.7	2.9	40.0
	5	5	13.5	14.3	54.3
	6	6	16.2	17.1	71.4
	7	2	5.4	5.7	77.1
	8	5	13.5	14.3	91.4
	9	3	8.1	8.6	100.0
	Total	35	94.6	100.0	
Missing	System	2	5.4		
Total		37	100.0		

Table 2 (b) Database competency summary

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
A_db_create	35	1	9	5.19	2.352
Valid N (listwise)	35				

Table 2(a) and 2(b) reveal interesting statistics, almost 50% students perceived themselves having “good” knowledge of ACCESS database. This would imply that almost half the class should have some table, querying and report building capabilities.

Test of Hypothesis

Three home works that have following requirements were used for this experiment:

- Access ORACLE software remotely
- Create table(s)
- Load data
- Answer queries

Students were provided handouts that described how to access ORACLE remotely and a sample database example was provided. SPSS software package was used to test the hypothesis. Table 3 summarizes the results.

Table 3: Software portability

Group Statistics

	A_db_create	N	Mean	Std. Deviation	Std. Error Mean
final	>= 5	20	82.2296	7.49611	1.67618
	< 5	12	63.8451	31.31699	9.04044

Independent Samples Test

	Levene's Test for Equality of Variances		t-test for Equality of Means		
	F	Sig.	t	df	Sig. (2-tailed)
final	21.178	.000	2.533	30	.017

Table 3 clearly shows that students that have average skills(>=5) in ACCESS do perform better in the database class than students that have less ACCESS skills. This would imply students can translate some skills from one software to another without instructor’s intervention.

DISCUSSION AND CONCLUSIONS

The results presented above are encouraging for instructors and has implication for counselors. For student counselors it would imply encouraging students to have some database software competency before enrolling in the database course. One of the student's complaint is the rigor of the course, and many students can not keep pace with it. It may also mean officially revising pre requisites for the database course by *requiring* some database competency before allowing students to enroll in the course. For instructors this would imply, they can concentrate on pedagogical issues of the course instead of spending too much time teaching the syntax of the software.

Results of this experiment must be interpreted with caution since they are based on one semester study. Our experiment needs validation. We intend to repeat this experiment in future semesters to validate our results.

REFERENCE

Bloom B. S. (1956). Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. New York: David McKay Co Inc

Garen, Ken (2007), Software Portability Weighting Options, making choices, The CPA Journal, available at: http://www.nysscpa.org/cpajournal/2007/1107/perspectives/p10.htm

Stern, Tom (2008), The Skills Portability Factor, available at: http://www.fastcompany.com/articles/2008/08/tom-stern-skill-portability.html