

Task Specific Programming Languages as a First Programming Language

Mark Guzdia
 W. Michael McCracken
 Allison Elliott
 College of Computing
 Georgia Institute of Technology
 Atlanta, GA 30332-0280

This research investigates whether there is a difference in the acquisition of programming skills and knowledge as a function of a student's first language. Our research is concerned with the comparison of task specific languages and general programming languages. In many engineering programs students are first exposed to the principles of computational solutions to problems by means of task specific languages, such as MatLab. They are then either expected to be able to use, or are specifically taught programming using more general purpose languages, such as C. Our question is whether there is a developmental preference for learning a task specific language first, or a general purpose language first. Historically, educators have emphasized fundamentals prior to application. A case could therefore be made that a student should be taught general programming skills in the context of a general purpose language before solving problems in a task specific language. More recently, contextualized educators would prefer the initial learning of task specific languages. Our research anticipates answering the question of the effectiveness of transfer of programming skills as a function of first language learning.

The dimensions of this question include but are not limited to, how the languages are used, what types of problems are presented to the students, is transfer prompted between the languages, do students look for surface or structural similarities, and what are the assumptions and expectations of the faculty who teach these languages. There are also some assumptions. We assume that some engineering students will be eventually writing programs using general purpose languages, and we assume that the skills of these students will be of the caliber of second year computer science students. In other words, the students will be able to construct relatively complex programs as solutions to real problems. We also assume that the faculty expect the students to be able to use programs to solve relatively complex problems rather than just as equation solvers and generators of nice graphs. Otherwise the research is moot.

We have developed a comparison of task specific languages and general purpose languages to allow us to investigate ontological boundaries between languages and their impact on transfer. For example, MatLab essentially has no typing. It uses built in types. Whereas, general purpose languages have various types, including enumerated types, and support the construction of complex data structures around those types. Does this difference cause a boundary to transfer? If a student learns MatLab first, will data types be more difficult to learn? If a student learns a general purpose language first, will they be able to transfer their skills to a language that prevents them from constructing many of the structures they have previously used?

Our initial work on this problem was in the form of a survey we sent to engineering educators who teach programming in their curricula. We specifically asked them whether they were able to see differences in the skills of the students as a function of the first and successor languages. The results of the survey were mixed, and the results showed no preferences for the first language. We view the survey as a starting point in our research and are using the results to more carefully guide our work. The important results from the survey were the comments received from the respondents, which have prompted us to clarify what we are looking for. For example, we are interested in students who utilize the full set of features of a task specific language, and not those who use them as equation solvers. Therefore, we are currently collecting data from a class at Georgia Tech to more carefully construct our questions and to gather some initial results on task specific languages as a first programming language. The class is an engineering class using MatLab as a first programming language. We have a clearer understanding of the specific knowledge to be acquired by the students in this class and have assessed their skills prior to the class.

Our results to date are inconclusive. We are concerned that the inconclusiveness of the results may lead some, including ourselves, to believe that the question is rhetorical. Rather than debate which should

come first, a more useful result may be specific guidance to faculty to prompt transfer from one language to another. We believe more effective measures of programming skill must also be used to determine what relative differences, if any, exist in the selection of the first language.

