Work in Progress: Student Perception of Computer Programming Within Engineering Education: An Investigation of Attitudes, Beliefs, and Behaviors

Dr. Kelly S Steelman, Michigan Technological University

Dr. Kelly Steelman is an Associate Professor in the Department of Cognitive and Learning Sciences and an Affiliated Associate Professor in the Department of Mechanical Engineering and Engineering Mechanics at Michigan Technological University.

Dr. Michelle E Jarvie-Eggart P.E., Michigan Technological University

Dr. Jarvie-Eggart is a registered professional engineer with over a decade of experience as an environmental engineer. She lectures in the Engineering Fundamentals department at Michigan Technological University. Her research interests include online learning, first year learners, sustainability and diversity in engineering.

Kay L Tislar, Michigan Technological University

Kay Tislar is a lecturer in psychology at Michigan Technological University.

Dr. Charles Wallace, Michigan Technological University

Dr. Charles Wallace studied Linguistics at the University of Pennsylvania and the University of California before earning his Ph.D. in Computer Science at the University of Michigan. He has been on the faculty of the Michigan Tech Computer Science Department since 2000. His experiences as a computer scientist, linguist, and software developer drive his research exploring how humans can better understand, build, and use software. His work has been funded by the National Science Foundation, Google, Microsoft Research, and the U.S. Department of Defense.

Dr. Wallace's Agile Communicators project, supported by an NSF IUSE award, seeks to build an enhanced curriculum for computing programs that emphasizes inquiry, critique and reflection, grounded in authentic software development settings. Tools in this project include process oriented guided inquiry learning, automated feedback to students through an intelligent tutoring system, case studies in software communication, and guided reflective exercises on team communication. As part of this research, the Agile Communicators team has investigated communication practices in a variety of student and professional software development environments.

Wallace has been intimately involved with undergraduate Computer Science curriculum development since his arrival in 2000. He cofounded Michigan Tech's Software Engineering degree program in 2003. Wallace currently serves as Director of Undergraduate Programs for the Computer Science Department. In conjunction with his research projects, he has founded local outreach efforts in computer education for middle and high school students and digital literacy for senior citizens.

Dr. Nathan D Manser, Michigan Technological University

Dr. Manser is a multidisciplinary engineer with over a decade of experience as a mining and environmental engineer. He lectures in the Geological and Mining Engineering and Sciences Department at Michigan Technological University. His research interests include engineering education, natural resource management, and sustainable engineering systems.

Mrs. Briana C Bettin, Michigan Technological University
Leo C Ureel II, Michigan Technological University

Leo C. Ureel II is a Senior Lecturer in Computer Science at Michigan Technological University. He has worked extensively in the field of educational software development. His research interests include intelligent learning environments, computer science education and software engineering. He currently has primary responsibility for the introductory programming courses at Michigan Tech.

Work in Progress: Student Perception of Computer Programming Within Engineering Education: An Investigation of Attitudes, Beliefs, and Behaviors

Computer programming has become a fundamental engineering competency, but engineering students outside of traditional computing-intensive specialties often fail to understand its value. In our quest to enhance student awareness of the importance of programming, we seek to understand the relationship between characteristics of individual learners (in particular, their position in the Fixed-*vs*.-Growth Mindset continuum, and their Openness to Experience) and their attitudes toward programming. We conducted a study involving engineering students in fields not traditionally associated with computing. Three surveys were administered, investigating the students' attitudes toward Computer Science, their Openness to Experience, and their Mindset (Fixed or Growth). The results indicate that interventions emphasizing the usefulness of programming, particularly across a broad range of applications, may have a positive effect on student attitudes, and that interventions focusing on algorithmic thinking may have a significant effect on student confidence and therefore intention to go further in learning programming.

Introduction

Although most engineering faculty and professionals view computer programming as an essential part of an undergraduate engineering curriculum, engineering students do not always share this viewpoint. In fact, engineering students outside of computer and electrical engineering may not realize the value of computer programming skills until after they have graduated and advanced in their career [1]. Failure to find value in computer programming may have negative consequences for learning. Indeed, engineering students who do not view programming as interesting or useful show poorer performance on tests of programming concepts than students who do [2]. This finding is consistent with theories of technology acceptance [3-4] that emphasize perceived usefulness as a key determinant of attitudes toward a technology and subsequent use or disuse of it. This suggests that engineering coursework should include specific interventions that emphasize the utility of programming skills for a career in engineering.

Intervention effectiveness, however, may depend in part on the characteristics of the individual learners, including their beliefs about the nature of intelligence and their Openness to new experience. Intrinsic learner characteristics, such as whether a student possesses a *Fixed* or *Growth Mindset,* may affect a students' overall academic success [5]. Students with a Fixed Mindset tend to view their intelligence and abilities within an area as unchangeable; their self-talk with regards to learning programing might include, "*I'm not good at programming*". In contrast, those with a Growth Mindset are more likely to view their intelligence and abilities as something that can be developed; their inner dialogue might include, "*I don't know how to program yet, but I can get better at programming*". These differences may shape student's attitudes toward learning a new skillset like programming and their intentions to develop this skill in the future.

Another personality trait that may affect learning programming is Openness to Experience. Also referred to as intellect, Openness is one of the Big 5 Personality Traits [6]. Individuals with this trait can be described as cognitive explorers; they are curious and excited about learning, and they

actively seek out cognitively challenging situations [7]. Recent work has found that Openness to Experience is significantly correlated (r=0.41) with scores on the Computational Thinking Test [8]. Other recent work has suggested that Openness may be increased through cognitive training [9].

The purpose of the current work is to understand engineering students' attitudes toward and experiences with computer programming, and to assess the relationship between their attitudes, Mindset, and Openness to Experience.

Methods

Participants. Seventy-nine engineering students participated in the study as part of a general education Psychology course. Of these, 73 participants (including 22 women) completed the full study and correctly answered 4 trap questions that were included to assess whether participants were paying attention during the study. The mean age of participants was 19.58 (SD= 0.91). Study participants reported the following engineering majors: Biomedical Engineering (11), Chemical Engineering (7), Civil Engineering (9), Environmental Engineering (2), General Engineering (1), Material Science and Engineering (4), and Mechanical Engineering (39). Students in computing-related engineering fields (like Software Engineering and Electrical Engineering) as well as students with computing-related minors were not included in the sample.

Materials and Procedure. Participants completed a computer language inventory and three surveys. The first survey inquired about students' attitudes toward Computer Science [10]. The scale was developed specifically to assess engineering student's attitudes toward Computer Science and contains 38 questions related to student's (1) confidence in their ability to learn Computer Science skills, (2) interest in Computer Science, (3) beliefs about the usefulness of learning Computer Science for their career goals, (4) beliefs about professionals in the Computer Science field, and (5) perceptions about Computer Science as a male field. Participants responded to each question on a four-point Likert-like scale ranging from *strongly disagree* to *strongly agree*. Questions from each of the five categories were averaged and summed to produce an overall measure of Attitudes toward Computer Science. High scores indicated positive attitudes.

The second survey posed questions related to different aspects of Openness to Experience [11]: intellectual efficiency, ingenuity, curiosity, aesthetics, tolerance, and depth. Participants responded to each question on a five-point Likert scale ranging from *strongly disagree* to *strongly agree.* Ratings on each question were summed to produce an overall Openness score. Additionally, the scores for intellectual efficiency, ingenuity, and curiosity were summed to produce an Intellectual Openness score. Scores for aesthetics, tolerance, and depth were combined to produce a Cultural Openness score.

Finally, the third survey probed participants' beliefs about the nature of intelligence and whether it is fixed or can be developed [5]. The survey contained four questions which were rated on a six-point Likert-like scale ranging from *strongly agree* to *strongly disagree*. Low scores indicated a Fixed Mindset, while higher scores indicated a Growth Mindset.

Results

The current analysis focuses on five self-reported beliefs and intentions about programming and their relationship to Mindset, Openness to Experience (Openness), and Attitudes toward Computer Science (ACS). Average responses to these five key questions are illustrated in Figure 1. Participants generally agreed with the statements that (1) programming is a useful skill for engineers, (2) that they could be a good programmer if they took some college-level classes, and (3) that knowing at least some basics about programming will help them find a job in the field. On average, participants slightly agreed with (4) the intention to take at least one programming course. Participants slightly disagreed with the statement that (5) the way programs have to be constructed makes no sense.

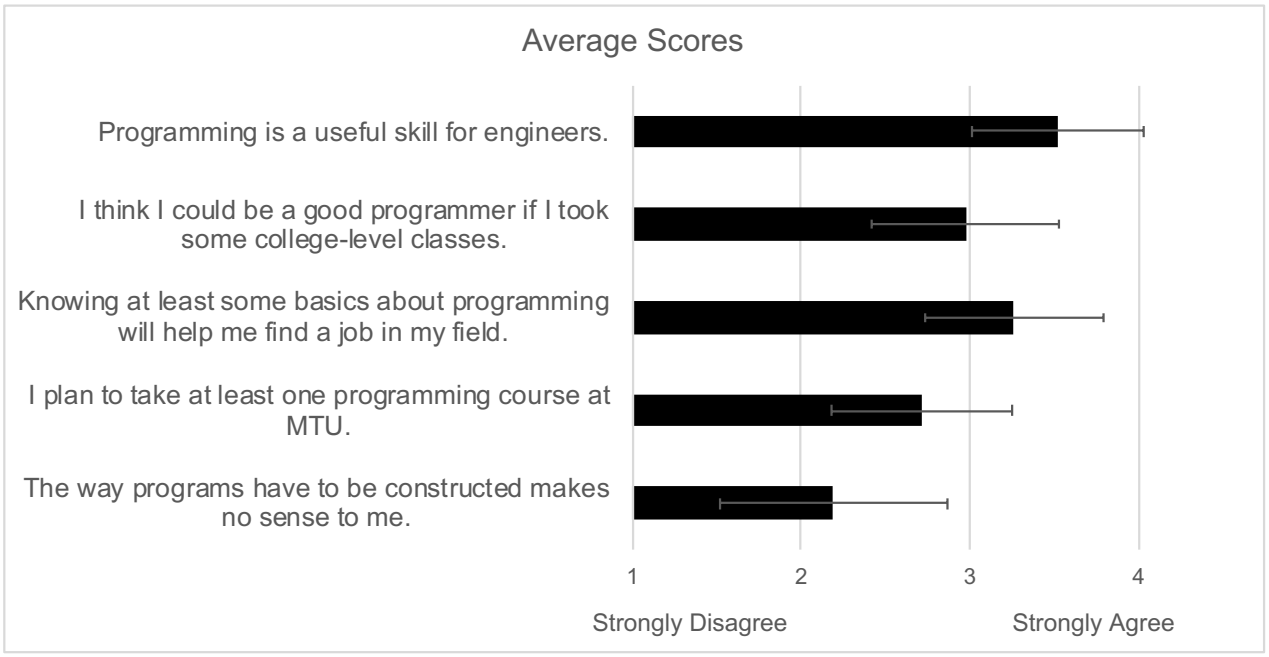

Figure 1. Average scores on key question. Average usefulness scores. Error bars represent the standard deviation of the mean.

Pearson Correlations among all of the variables were calculated and are presented in Table 1. Significant correlations ($p<.01$) are marked with asterisks. Scores on the ACS were positively correlated with the view of programing as a useful skill and the intention to take a programming class. In fact, of all variables we examined, intention to take at least one future programing course was most strongly correlated with the ACS score. Examination of the ACS subscales (Table 2) provides more insight into this relationship. Looking at the five underlying constructs, only confidence, interest, and usefulness to one's career were significantly correlated with intention to take a future course.

ACS was also negatively correlated with the view that the way programs have to be constructed does not make sense. Examining the ACS subscales, responses to the statement that program construction makes no sense was strongly negatively correlated with both confidence and interest in Computer Science. Importantly, although participants only mildly disagreed with the statement that program construction does not make sense, ratings on this question were negatively correlated

with intention to take at least one programming class, suggesting that individuals who struggle with algorithmic thinking are less likely to plan to take programming courses in the future.

Table 1

*Correlations among mindset, Openness, ATC, and key programming questions. **p<.01*

Variable	1	2	3	4	5	6	7	8
1. Mindset	–							
2. Openness to Experience	**0.36****	–						
3. Attitude toward CS	0.12	**0.41****	–					
4. Programming is a useful skill for engineers.	0.06	0.13	**0.34****	–				
5. I think I could be a good programmer if I took some college-level classes.	-0.03	0.28	0.29	0.1	–			
6. The way programs have to be constructed makes no sense to me.	0.25	-0.17	**-0.33****	0.03	-0.39	–		
7. I plan to take at least one programming course at MTU.	-0.04	0.09	**0.40****	0.15	0.30	**-0.30****	–	
8. Knowing at least some basics about programming will help me find a job in my field.	0.10	0.09	0.28	0.22	0.12	0.01	0.12	–

Table 2

*Correlations between ATC underlying constructs and key programming questions. **p<.01*

Variable	Confidence	Interest	Gender	Usefulness	Professional	ATCS
I think I could be a good programmer if I took some college level classes.	**0.42****	0.27	0.00	0.10	0.12	0.29
The way programs have to be constructed makes no sense to me.	**-0.62****	**-.40****	0.02	-0.01	-0.04	**-0.33****
I plan to take at least one programming course at MTU.	**0.31****	**0.33****	0.13	**0.37****	0.16	**0.40****
Knowing at least some basics about programming will help me find a job in my field.	0.10	0.10	0.41**	**0.40****	-0.02	0.28

Table 3

*Correlations between the intellectual and cultural dimensions of Openness and key programming questions. **p<.01*

Variable	Intellectual Openness	Cultural Openness
Attitude toward CS	**0.37****	0.29
I think I could be a good programmer if I took some college-level classes.	**0.41****	0.08
The way programs have to be constructed makes no sense to me.	**-0.34****	0.01
I plan to take at least one programming course at MTU.	0.15	0.01
Knowing at least some basics about programming will help me find a job in my field.	0.05	0.09
Programming is a usefull skill for…		
engineers.	0.12	0.09
scientists.	0.21	0.21
educators.	0.14	**0.41****
people in social science fields.	0.11	**0.29**
people in the medical field.	0.09	**0.39****
writers.	-0.06	**0.37****

As shown in Table 1, Openness to Experience was positively correlated with both mindset and ACS, but was unrelated to all other factors. Analysis of the two underlying dimensions of Openness — Intellectual Openness and Cultural Openness — revealed additional insights. As shown in Table 3, Intellectual Openness was significantly positively correlated with ACS as well the belief that one could be a good programmer if they took some college-level classes. It was also negatively correlated with the view that the way programs have to be constructed does not make sense.

Cultural Openness, in contrast, was not significantly correlated with ACS nor with intentions to take a programming class, but it was correlated with the view that programming was a useful skill in other non-engineering disciplines. (As illustrated in Figure 2 below, participants viewed programming as most useful for engineers and scientists, with average ratings for other disciplines falling somewhere between "disagree" and "agree"). Although the reason for this relationship is unclear, it may suggest that individuals who are more open-minded have an easier time speculating about the utility of programming for people in disciplines that students may view as having little in common with their own. Another possibility is that students who are more culturally open may have a broader range of academic experiences that may have allowed them to see beyond stereotypes that certain skillsets are relevant to only STEM or non-STEM fields.

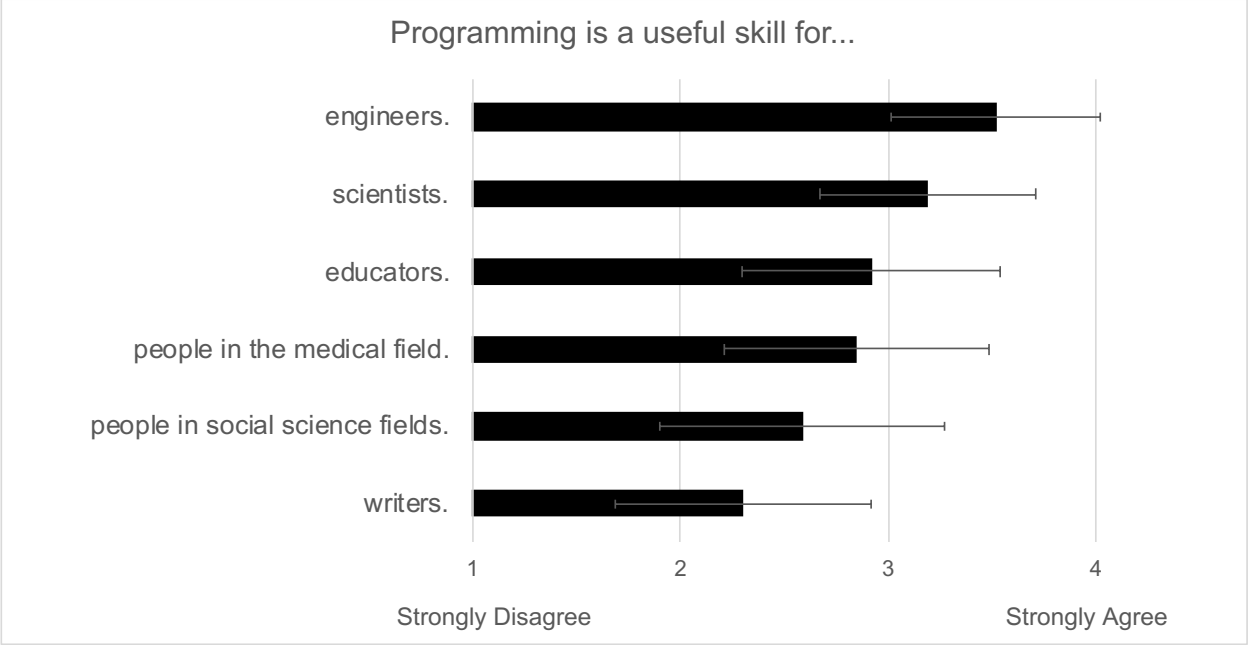

Figure 2. Average usefulness scores. Error bars represent the standard deviation of the mean.

Discussion

The current results emphasize the importance of perceived usefulness of programming on intention to take programming classes in college. This finding is consistent with the Technology Acceptance Model (TAM), which has been widely used within the field of information technology to predict intention to use technologies [3]. Within the TAM, behavioral intention to use a technology is directly influenced by both the perceived usefulness and the perceived ease of use of the

technology. Perceived usefulness, in turn, is directly affected by several other factors, including the degree to which the technology is considered applicable to one's career path [12].

In the fall of 2019, our research group implemented pilot interventions focused on increasing the perception of the usefulness of programming for engineers. The intervention comprised presentations from industry professionals explaining the usefulness of programming within their careers, how they overcame programming challenges, and how the skill of programming contributed to their success. Although the current study results further motivate the focus of these interventions, thus far the interventions have proven unsuccessful at producing statistically significant changes in students' acceptance of programming, possibly due to the fact that they were one-time, online presentations that did not afford students with the opportunity to ask questions nor require them to reflect upon their own career plans. One unexpected finding in the current study was that students who scored higher on the Cultural Openness dimension were more likely to view programming as useful in other fields. This suggests that it may be useful to focus interventions on increasing a sense of the broad utility of programming across all disciplines.

Another useful finding was that intention to take a future programming class was negatively correlated with ratings on the statement, *"The way programs have to be constructed makes no sense to me."* Agreement with this statement indicates difficulty with algorithmic thinking, suggesting that interventions focused on algorithmic thinking may be effective at increasing confidence and interest in computer science and increasing intentions to develop programming skills while in college.

Of course, the data presented here is correlational. It is possible that the relationships we have discussed reflect effects of other underlying variables that we have not yet assessed. Nevertheless, the current data point to areas of potential focus for future interventions. A further caveat is that the current work assesses only students' intentions to take a future programming classes and their belief that if they do so that they have the potential to become a good programmer. It is possible that focusing on algorithmic thinking and perceived usefulness may increase intention, but may not ultimately yield the desired behavior. The Theory of Planned Behavior (TPB), however, suggests that positive attitudes and behavioral intention are two key predictors of future behavior [13]. The TPB has been used to inform the design of successful outreach programs aimed at recruiting more under-represented minorities into engineering [14] and to study entrepreneurial intention among science and technology students [15]. TPB has also been used to predict STEM major and career choice for high school juniors and seniors, with attitude and intention being the most predictive components of the model [16]. Together, these programs document the importance of attitudes driving behavioral intention and action. TPB further identifies subjective norms, social norms, and perceived behavior as additional factors that affect whether intentions lead to actions. Our future interventions, therefore, should also consider establishing norms surrounding programming as an expected skillset for engineers early in the engineering curriculum.

Conclusion

This project is part of a larger research program aimed at understanding the factors that influence engineering students' adoption of technology and the development of skillsets that will enable future engineers to keep up with the rapid pace of technological change. The results suggest that effective interventions should focus on the development of algorithmic thinking and reinforce the

utility of programming as a skill, both generally and specifically within engineering careers. We have repeated this study with another large cohort of students, which will allow us to confirm these results and also examine differences between engineering and computing students. As noted earlier, we recently piloted a one-time video-based intervention. As it did not produce shifts in attitudes toward programming, our future work will focus on addressing the usefulness of programming throughout the first year Engineering curriculum and providing opportunities for students to reflect upon the relevance of programming to their own career goals. Our intervention will be further informed by recent focus groups conducted with first year and fourth year engineering students that examined how students learned various technologies (e.g., solid modeling software, Matlab) and their expectations about learning new technologies in the workplace.

References

[1] Sterian, A., Dunne, B., & Blauch, A. (2005, June), The case for computer programming instruction for all engineering disciplines. Paper presented at the 2005 Annual Conference, Portland, Oregon. https://peer.asee.org/15520

[2] Lingar, L., Abbey Williams, D., McCord, R., & Jeldes, I. (2017). Work in progress. Motivation profiles of non-major computer programmers in a flipped classroom environment. In Proceedings of First Year Engineering Experience (FYEE) Conference, American Society for Engineering Education.

[3] Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS quarterly, 319-340.

[4] Venkatesh,V., Morris, M., Davis, G., Davis, F. (2003). User Acceptance of Information Technology: Toward a Unified View. MIS Quarterly, 27 (3), 425-478.

[5] Dweck, C. (1999). Self-Theories: Their role in motivation, personality, and development. Philadelphia: Psychology Press.

[6] John, O. P., Naumann, L. P., & Soto, C. J. (2008). Paradigm shift to the integrative big five trait taxonomy. Handbook of personality: Theory and research, 3(2), 114-158.

[7] McCrae RR, Costa PT., Jr . The five-factor theory of personality. In: John OP, Robins RW, Pervin LA, editors. Handbook of personality theory and research. New York, NY: The Guilford Press; 2008. pp. 159–181.

[8] Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior, 80*, 441-459.

[9] Jackson, J. J., Hill, P. L., Payne, B. R., Roberts, B. W., & Stine-Morrow, E. A. (2012). Can an old dog learn (and want to experience) new tricks? Cognitive training increases Openness to Experience in older adults. Psychology and aging, 27(2), 286.

[10]Hoegh, A. and Moskal, B. M. (2009). Examining science and engineering students' attitudes toward computer science. In FIE'09 Proceedings of the 39th IEEE International Conference on Frontiers in Education Conference, 1306-1311, IEEE Press.

[11] Woo, S. E, Chernyshenko, O. S., Longley, A., Zhang, Zhi-Xue, Chiu, Chi-Yue, & Stark, S. E. (2014). Openness to experience: Its lower level structure, measurement, and cross-cultural equivalence, Journal of Personality Assessment, 96(1), 29-45, DOI: 10.1080/00223891.2013.806328.

[12] Vankatesh, V., and F. D. Davis. (2000). A Theoretical Extension of the Technology Acceptance Model: Four longitudinal field studies. Management Science, 46(2), 186.

[13] Ajzen, I. (1985). From intentions to actions: A theory of planned behavior. In J. Kuhl & J. Beckmann (Eds.), Action-control: From cognition to behavior (pp. 1 l-39). Heidelberg: Springer.

[14] Kuyath, S. (2005), *Applying The Theory Of Planned Behavior: Recruiting Underrepresented Minorities To Engineering And Engineering Technology* Paper presented at 2005 Annual Conference, Portland, Oregon, June 12- 15. https://peer.asee.org/14240

[15] Roy, R., Akhtar, F. & Das, N. Entrepreneurial intention among science & technology students in India: extending the theory of planned behavior. Int Entrep Manag J 13, 1013–1041 (2017). https://doi.org/10.1007/s11365-017-0434-y

[16] Moore, M. and J. Buurus, (2018). Predicting STEM Major and Career Intentions with the Theory of Planned Behavior. The Career Development Quarterly. June 2019, Vol. 76. 139-155.