Impact of Proficiency in English on the Intuitive Understanding of Computer Science Concepts

Ismat Aldmour, AlBaha University, Saudi Arabia

The European Conference on Education 2022
Official Conference Proceedings

Abstract
Computer science terms like: Code, Analysis, Protocol, Encapsulation, Validation, Sampling, Model and many more are borrowed from English with their meanings slightly altered to suite computer science. This makes initial computer science acquiring more difficult for non-native English students, while it is facilitated for students of higher English proficiency. This is sort of a transfer from language proficiency to computer science which is similar to the known concept of transfer from one language to another in new language acquisition. The paper presents a test for assessing this transfer by investigating students' understanding of selected terms in both technical and non-technical contexts. The terms were selected to represent computer science sub-concepts as defined in the literature; hence, students' understanding of these terms in everyday non-technical uses, measures potential students' understanding of these same terms in computer science technical uses. The test was applied on Arabic speaking students of different English proficiency and different maturity levels. It was found that the intuitive understanding of the terms in computer science improves with improved English, but no impact of maturity was found. Computer science students' records revealed an association between computer science learning and English level which is attributed partially to this transfer.

Keywords: Computer Science Education, Language and Education, Language Transfer

1. Introduction

Al-Nasser (2015) describes the outcomes of English language acquisition of school leavers in Saudi Arabia: "after studying English for about 9 years, school leavers are, in most cases, unable to speak or write a single flawless sentence in English.". Meanwhile, in Albaha University in Saudi Arabia, English is the language used to teach computer science (CS). Therefore, this presents a challenge to inducing an effective teaching-learning process in CS.

Interrelationship between English proficiency and computer science learning is an underdeveloped problem in literature. Most research focus on programming difficulties and the errors committed by novice programmers attributed to language in addition to other factors of math and algebra (Clancy, 2004; Ebrahimi, 1994; Jackson, Cobb, & Carver, 2005; Miller, 2014, 2016).

In a previous paper (Aldmour & Nylen, 2014), it was proposed that proficiency in English and the corresponding culture can lead to an intuitive initial understanding of CS terms and concepts. It was a work in progress paper which proposed also the methodology used here to test this dependence. The actual testing was conducted at later times with results presented in this paper.

The approach here is to look at the problem as a transfer from language (English) to CS that is in favor of native English speakers but may impede the learning of non-native English speakers. This is similar to the transfer occurring from the knowledge of one language (L1) to the learning of a second language (L2) in second language acquisition (SLA) concept in linguistics. A test is designed to measure transfer from English to CS similar to testing transfer from L1 to L2 in SLA. The test was applied to native Arab students of different proficiency levels in English. Test results were analyzed to investigate if a connection between the proficiency in English and the intuitive understanding of computer science terms and corresponding concepts exists. The connection (if existing) will be of great value as it sheds light on how non-native English students initially understand computer science terms, which may contribute to learning the concepts and the related CS tasks.

Additionally, knowing that a connection like this exists, it can provide basis for designing English courses for a specific purpose of enhancing learning and teaching in CS of non-native English speakers. This is similar to the recommendations in (Nation, 2003) about the importance of communicating meaning and respecting the role of the first language in foreign language learning.

2. Background

2.1 Culture, Language and Learning

Most important forms of human cognitive activity develop through interaction within social and material environments, including conditions found in instructional settings (Burgstahler, 2011). Lee (2005) assures that "Learning is enhanced—indeed, made possible—when it occurs in contexts that are culturally, linguistically, and cognitively meaningful and relevant to students". For students studying in another language with the associated culture, home languages and cultures encompass the tools that students use to construct their understandings of the world. "L1 provides a familiar and effective way of quickly getting to grips with the meaning and content of what needs to be used in the L2" (Nation, 2003). Many research works on students learning CS or other disciplines (Tobin & McRobbie, 1996; Tenenberg & Knobelsdorf, 2014; Lee, 2005) assert the role of students' cultural-environment and prior linguistic knowledge factors for limited English proficiency learners. More specifically, in the domain of CS teaching, Zendler, Spannagel, & Klaudt (2011) say that computer science curricula must not be based on fashions and trends, but on contents and processes that are, among other factors, related to everyday language and/or thinking.

Hence, any previous conceptual understanding gained using mother's language and culture, e.g. Arabic, will have to be recalled even if the language medium is different, e.g. English. This is especially true for those concepts that are linguistically or culturally related to science concepts and terms. Therefore, it is natural to conclude that learning can be more effective if the learners were developed or trained to obtain everyday English language and cultural meanings in a way similar to native English learners; i.e. they become more fluent in English and its culture.

Naturally, the terms and concepts of high importance in this mechanism will be those contributing toward learning the basic concepts of the discipline studied. Henceforth, central (basic) concepts in CS are next discussed.

2.2 Central Concepts in CS

Zendler and Spannagel (2008) determined the basic concepts in CS by surveying CS experts' opinions of what concepts are CS centrals. This results in a catalogue which classifies the central concepts in CS into 15 central concepts. These are namely: problem, data, computer, test, algorithm, process, system, information, language, communication, software, program, computation, structure, and model.

Zendler and Spannagel central concepts are of a wide nature spanning across the different CS subjects. They also represents what to be acquired by the students (e.g. upon completing the curriculum). Zendler and Spannagel also recommended that the central concepts have to be specified in more detail; i.e. subconcepts. For example, course specific concepts, called concept inventories (CI), to be used to asses gain on course level are obtained in (Goldman et al., 2010) for three introductory computing subjects: discrete mathematics, programming fundamentals, and logic design. In obtaining the CIs, they also followed an empirical approach that is based on Delphi process for collecting information and reaching consensus in a group of experts.

Hence, computer terms (to be used in the test described later in the paper) are selected to represent subconcents (or concept inventories) which can be classified under the 15 central concepts. Moreover, any of the terms selected has to represent a similar concept (have similar meanings) outside the CS discipline in everyday language. Understanding these terms in a computing context is therefore a measure of understanding of the wider central CS concepts.

This relation between language and CS terms and concepts is further exemplified by making reference to the second language acquisition concept and related language transfer phenomena as outlined in the coming section.

2.3 Language (to language) transfer and Language to CS transfer

The influence of English on learning computer science can, somehow, be looked at as if we look at the impact of knowing one language (first one) on the learning of a second language in what is called second language acquisition (SLA). In this analogy, the first language is English (as it is the medium of instruction) while the second language is computer science. In SLA, language transfer occurs when the second (new) language learning is influenced by the previously known language(s) (Zhai, 2012). Transfer is defined in (Ringbom & Jarvis, 2009) as the learners' reliance on perceived and assumed cross-linguistic similarities and it can be manifested at three different levels: item transfer (e.g., sound, morpheme, word, phrase), system transfer or procedural transfer (word forms and order), and overall transfer.

Effect of L1 on L2 learning is limited when considering L1 formal features, but it will be pervasive when considering L1 meanings (Lantolf, Thorne, & Poehner, 2015). This is in line with our work as it emphasizes the effect of meanings.

Computer science courses are mainly written in English and they utilize the readers' familiarity with the corresponding culture. Computer science terms generally originate in English and many computer science terms and concepts are English words for more or less similar everyday's phenomena. Examples are: handshaking, protocol, procedure, syntax, validation and piggybacking. A CS student who first encounters such terms in his CS studies can immediately build an intuitive initial understanding of their possible meanings and usages in the discipline provided that he/she is aware of their everyday English meanings and usages.

As language exists before CS, we view this relevance, as a kind of transfer. Let this transfer be denoted as Language (English) to Computer Science (L-CS) transfer. In this paper, we seek to investigate whether the students' proficiency in English and its culture influence their intuitive understanding of CS terms and concepts.

3. The Test

3.1. The Role of the Test and its Basis

In this section, we proceed to the test. It is designed to test students' intuitive understanding of selected terms in computer science and to correlate this with their knowledge of their corresponding meanings and usages in everyday English. The selected terms are assured to represent subconcepts in CS; hence, knowledge of their meanings in CS context is some indicator of intuitive understanding of CS concepts. Also, knowledge of their English meanings is some indicator of the students' proficiency level in English. Both levels are only indicators and are not meant to be professional English and/or CS proficiency level tests.

We argue that if any intuitive understanding of CS concepts existed in students subjected to the test, it is attributed to the impact of English on CS; i.e. English is the causal factor in this association. Consequently, the test is to be used only on students with no knowledge in computer science, so that the understanding they may show on the test is an intuitive one attributed to their understanding of the terms based on their language background. Also, other factors impacting students' knowledge of CS concepts, beside their English level, are isolated (as described later). The test is not to find about the impact on CS learning or on the final understanding of CS concepts. After all, other factors; e.g. enhanced pedagogy, may overcome initial difficulties.

Using language transfer concept terms from linguistics, the test will detect whether knowledge of English results in a positive transfer to Saudi students' ability to get meanings from unfamiliar computer science terms and concepts. Specifically, it tests the ability to infer meanings and usages of terms which are of bi-use in both everyday English and in computer science.

3.2. Characteristics of the Test Groups

In our test, we have test groups to compare that have different English proficiency levels but can be roughly considered equivalent on all other factors. Also, as other factors of age and the extent of previous studies have an impact on language transfer in second language acquisition (Chamot, 2004; Nikolov & Djigunović, 2006; Raheem, 2018; Zhai, 2012), it is reasonable to assume that maturity is also a factor in L-CS transfer. Students of the same level of study are considered of the same maturity level (as level of study combines both age and the extent of previous studies).

Groups of Saudi students at two different levels of maturity and at two different levels of proficiency in English, all with little knowledge in computer science, are compared. Basically, these groups were drawn from different majors and grades at the university, henceforth, with regard to maturity and English level we expect that each group which is of the same study area and the same level of study to be homogeneous with regard to maturity and English level.

However, individuals in any group may perform differently on the test due to other factors out of our control. Examples of these factors that we thought about are students with special extracurricular training in computer science, students with special past experience, e.g. students who lived outside the country for some period and students who received special English training different to others. Those students were excluded from the test. Also, university *GPA* and secondary school average (*SSA*) might contribute to an underlying aptitude towards both English and computing concepts; i.e. confounding variables. Hence, Efforts were exerted as well in order to rule out the effect of such confounding variables.

3.3. The Test Parts

The test is composed of two parts. In the first part, the students are asked to provide general information about themselves such as age, field of study, level of study, status of study, year of enrollment, accomplished credit hours so far and *GPA* (or *SSA*). As well, students who changed major, repeating students, students who are over aged, and students of exceptional *GPA*, were all pinpointed and excluded. Students were also asked to assess the level of their knowledge in computer science (novice user, intermediate, and programmer) and to provide information regarding any special training courses on computers, IT, programming, CS and English, and whether they had been abroad for some prolonged time. We used this information to exclude the test results of students who appear to be different to the rest of the group in a way which may impact their classification as students of a certain homogeneous group. Moreover, no group is created with student studying CS or a related area.

The second part of the test assesses the students' understanding of English words, both in their everyday use and in their use as computer science terms. Since the students are not expected to have prior knowledge in computer science, they are asked to use their intuitive understanding of the word to infer the computer science term meaning(s). For this part, 30 terms in CS are selected. Each one of them is classified as a subconcept under one of the 15 CS central concepts. The terms that are used in the test are purposefully chosen to be of bi-use in nature. For example, the term Syntax means, in everyday language, "The order, vocabulary and rules in which the words forming sentences and phrases, in human languages, come". In computer science the term Syntax is linked with the basic concept of Language whereby particular programming language syntax refers to order, spelling and rules with which the vocabulary, symbols and variables must have in a program depending on the programming language itself. Hence, ingredients and role of Syntax in English are almost the same ingredients and role of syntax in CS programming languages. It is therefore expected that students aware of Syntax meaning(s) in English will be able to infer its extended meaning(s) within computer languages concept.

#	CS Concept	CS Term	#	CS Concept	CS Term
1	Problem	---	9	Language	Procedural, Syntax
2	Data	Abstraction, Encapsulation, Representation, Validation, Verification	10	Communication	Protocol
3	Computer	Logic, Proxy	11	Software	Analysis, Interface
4	Test	Reliability	12	Program	Code, Control
5	Algorithm	Aloha, Piggybacking, Round-Robin	13	Computation	---
6	Process	Batch, Bootstrapping, Pipelining, Sampling	14	Structure	Substrate
7	System	Combinational, Embedded, Timesharing	15	Model	Imperative, Simulation, Synthesis
8	Information	Non-repudiation			

Table1: CS terms used in the test and the concepts they are linked to.

Aldmour & Nylen (2014) initially selected some terms, classified them based on their experience, and asked a number of colleagues in the field to review the initially selected terms and their classification and to suggest other terms that they may find more appropriate as subconcepts. A list is finalized as shown in Table 1 (Aldmour & Nylen, 2014). The table lists 30 terms together with the concepts that they are linked with, e.g. Code is linked with the Program concept, Abstraction with Data and Syntax with Language.

The second part of the test contains 30 test questions, one for each term. In any question, the student is given the first translation shown by Google Translate (translate. Google. com). The translation is given because we assume that the student will use some kind of quick translator in his studies. Each question is composed of a question sentence giving the term (and its translation) followed by two columns. Left column lists four options of everyday meanings, two of them are correct. Right column also lists four options of CS meanings with two options correct as well. The different answers are also given in Arabic as we target to test understanding only.

Table 2 shows how the term Syntax appears in the test (in English) as an example (correct answers given). The students were instructed to tick two correct answers from the four options in each column. The test answers are marked and analyzed as described below.

Choose the 2 correct meanings (left col.) that comes to mind for the term "Syntax" (التركيب) in normal English.		**Choose the 2 correct meanings (right col.) you expect for the term "Syntax" in computer science**	
Meanings and vocabulary connotations which form sentences and phrases	☐	Vocabulary, symbols and variables have particular order which depends on the programming language	☑
The order in which the words forming sentences and phrases come.	☑	Software lines define processes implemented by the computer needs to know	☐
Paragraphs consist of words and words consist of letters	☑	Each programming language has basic vocabulary set that the programmer has to know.	☐
The vocabulary of different languages	☐	The program consists of lines and the lines of words, symbols and variables.	☑

Table.2: Example of a question from the test.

4. Applying and Analyzing the Test Results

The test was applied on students of Albaha University in Saudi Arabia. It was first applied on a small group of students for validity and stability purposes and to ensure that the level of difficulty is appropriate. After adjustments, the test was applied to three different groups of students defined as follows:

- Group 1: This group is the group of high maturity and low English level (HL Group). The group consists of final year students in an area other than English and computer science, who study mainly in Arabic. Students in this group were selected to be Year 4 students with Arabic literature as their major.

- Group 2: This group is the group of both high maturity and high English level (HH Group). The group members were selected from Year 4 English literature and Year 4 business students who study mainly in English.

- Group 3: a group of first year students who only know English as a second language at secondary school level. This group represents the low maturity and low English (LL Group) level students.

Students in Group 2 (HH) are expected to have significantly more knowledge of English (Proficient level) than the students in groups 1 and 3 (Primitive level). Hence, any significant difference the test reveals between group 1 (HL) and group 2 (HH) that is favoring group 2 could be attributed to their higher level of English and could be an evidence of L-CS transfer.

Also, students in group 1 (HL) and group 2 (HH), the two senior level groups different only in English proficiency, are expected to be more mature than students of group 3 (LL Group of

Year 1). The purpose of testing group 3 is to compare their results to those of group 1 to be able to investigate the impact of maturity on L-CS transfer.

To quantify the transfer, the following two measures are first defined:

- Every Day English Proficiency (*EDEP*: score out of 60): Calculated as the sum score of the scores of the 30 terms on the left column (Everyday English meanings) with every term has 2 correct answers, hence, scored 0,1,or 2.

- Computer Science Concepts Understanding (*CSCU*: score out of 60): Calculated as the sum score of the scores of the 30 terms on the right column (CS meanings) with every term also has two correct answers, hence, scored 0,1,or 2.

A positive measure of transfer (*TR*) is then defined as the correct *CSCU* per correct *EDEP* per question (test item).

Notice that we elected not to classify the occurrence of negative transfer (if any) as negative transfer, or interference. This is following (Ringbom & Jarvis, 2009) in their linguistic study where they elected to describe this occurrence as the absence of relevant concrete (positive) transfer.

Table 3 shows how we defined the *TR* measure. In the table, the extra correct *CSCU* answers compared to *EDEP* answers (Cases A and D) are attributed to randomness, hence, they are not considered a transfer. Conversely, in cases E and F, *EDEP* is greater than *CSCU*, hence, there is no transfer in Case E (*TR*=0, students scored 2 correct in *EDEP* resulted in no correct CSCU) or partial transfer (*TR*=1) occurred only in case F (2 resulted in 1 only). The Hit (Case G), represents the case of a student knowing the two everyday language meanings (two correct *EDEP*) resulting in two corresponding correct *CSCU* answers.

Case	***EDEP***	***CSCU***	***TR***	Note
A	0	0, 1, 2	0	No transfer
B	1	0	0	No transfer
C	1	1	1	Transfer of 1
D	1	2	1	Transfer of 1
E	2	0	0	No transfer
F	2	1	1	Transfer of 1
G	2	2	2	Hit (transfer of 2)

Table.3: Positive transfer (TR) definition.

To assess the results of each group in inferring correct CS concepts from their English knowledge, correlation values between students' scores on *CSCU* and students' scores on *EDEP* for each group are obtained. With all the factors isolated as described above, we may assume causality; i.e. the proficiency in English is the causal factor for understanding the terms in the computing context. Hence, positive correlation result indicates that the students score higher in CS when they are more proficient in English. Pearson correlation factor is calculated for each group. The strength of the correlation, hence, the L-CS transfer level, is assessed at 5% significance level. Transfer occurs when the correlation is significant. The square of the correlation values indicates weaker relationships as the correlation values approach zero.

A null hypothesis is made that *CSCU* scores are not related to their *EDEP* scores for each and every one of the three groups.

Group	**Group Label**	**Description**	**Number of Students**
Group 1 (ARA 4)	HL	High maturity/ Low English (Arabic Literature Students - Y4)	32
Group 2 (BUS 4+ENG 4)	HH	High maturity/ High English Y4(*)	41
Group 3 (PREP)	LL	Low maturity/ Low English Y1 Preparatory (Y1) Students	23
(*) Group 2 is formed from 23 English Students and 18 Business Students Y4			

Table.4: Details of student groups.

Group	**Correlation *R***	**R^2**	**Significance *P***
Group 1 (ARA4)	0.2166	0.0469.	0.320857
Group 2 (BUS4+ENG4)	0.4809	0.2313	0.007142
Group 3 (PREP)	0.2233	0.0499	0.283289

Table.5: A summary of the correlation results.

5. Results and Discussion

Table 4 shows the students' groups tested and the number of students in each group. Table 5 shows the correlation measure results *R*, the square of correlation, R^2, and the significance of the correlation results *P*. From this table we find that R^2 values are close to zero for groups 1 and 3, which indicates a weak relationship between *CSCU* and *EDEP*. Group 2 correlation result, as the significance level assures, is the only result of significance at $P<5\%$.

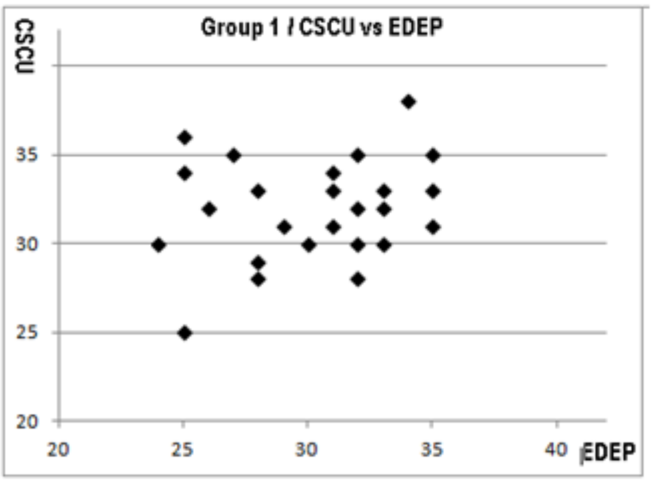

Fig. 1: *CSCU* versus EDEP for Group 1, ARA 4 (HL).

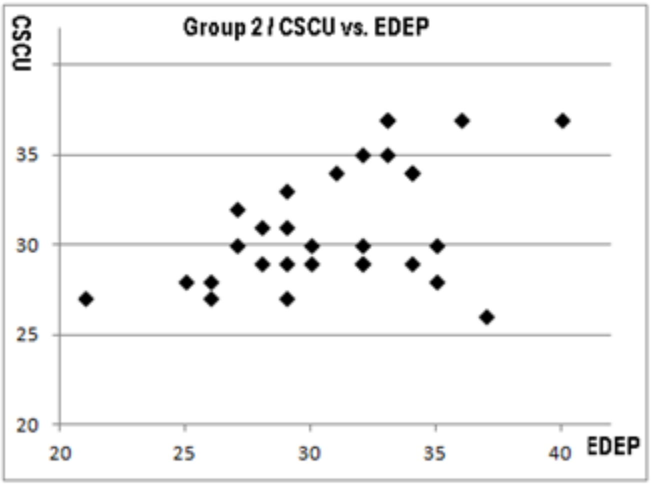

Fig. 2: *CSCU* versus *EDEP* for Group 2, Bus.4+Eng.4 (HH).

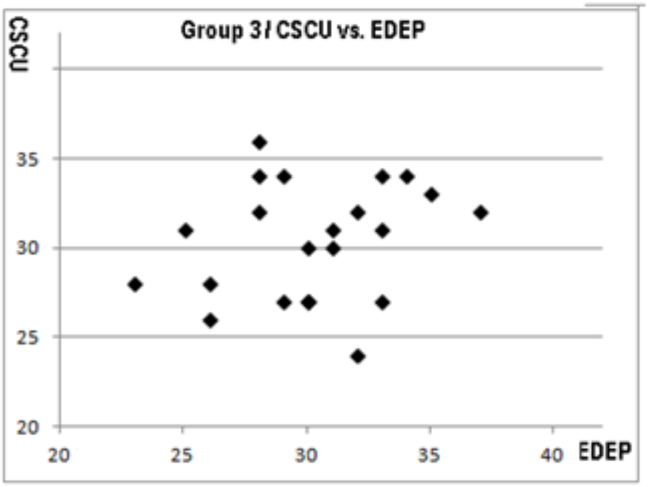

Fig. 3: *CSCU* versus *EDEP* for Group 3, Prep. Y1 (LL).

Figure 1, Figure 2 and Figure 3 show scattered plots of *CSCU* versus *EDEP* for groups 1, 2 and 3 respectively from which we see that a clear correlation pattern exists only for group 2.

The average transfer *TR* per student per term for group 2, the group with successful result, is calculated from the raw data and is found to be 0.38.

Hence, Group 2 students, the high English level students, are significantly more successful in inferring the meanings than the other two groups. That is, being knowledgeable in English makes acquiring computer science concepts easier. For both final year students studying mainly in Arabic (Group 1) and first year students (Group 3) the correlation results were insignificant with small R^2 values (weak relationship). Hence, we conclude that maturity has no impact on L-CS transfer and the results for Group 2 can be only attributed to English.

However, one might inquire whether this necessarily imply that this group (if were to study CS) would be more successful in the final learning of CS; i.e. the learning assessed by performance on a programming project, a term/end of term test, or the overall GPA.

Moreover, this study did not object to tell about the amount of direct correlation between proficiency in English and the final performance on CS as many other factors can take place.

Nevertheless, it is legitimate to extrapolate the above result of positive impact on the initial acquiring of CS and to expect positive impact on the final CS learning. However, no or little impact is expected when extra pedagogy or other measures stand for the weakness in English.

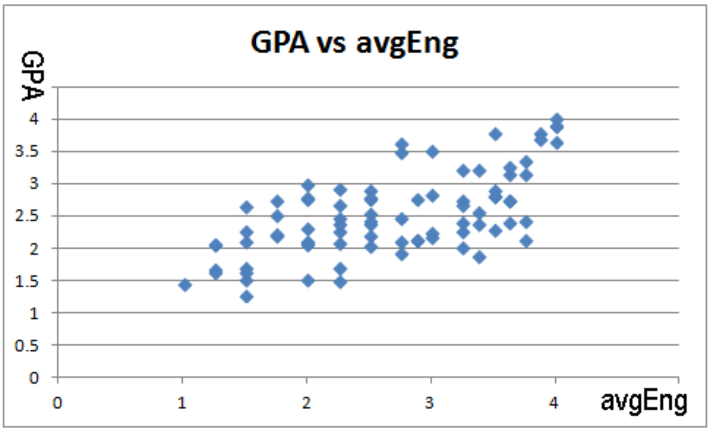

Fig. 4: *GPA* vs. *avgEng* (Average English Score).

Regardless of above argument, a statistical analysis using the records of third and fourth year CS students was done to find the correlation between CS students' proficiency in English (English 1 and English 2 grades average called *avgEng*) and their overall performance (*GPA*). This kind of analysis is an ex-post facto, non-experimental approach followed in many research works, e.g. (Martirosyan, Hwang, & Wanjohi, 2015). Figure 4 depicts the results obtained on a scattered plot of GPA versus *avgEng* scores of 82 CS students. A correlation value of 0.645 is calculated. This is a moderate positive correlation, which means that there is a tendency that a student *GPA* score goes up whenever his *avgEng* score goes higher. Again, we cannot attribute this correlation totally to our suggested transfer mechanism of English to CS concepts.

6. Conclusions

The paper reports on a test of students' comprehension of terms in both an English-language context and a computing context. The results showed positive correlation between proficiency of English and intuitive understanding of CS concepts (limited by the terms used in the test). In other words; improved English enhances initial acquisition of CS concepts. No impact of maturity on this relationship was found. This enhanced acquisition of CS is attributed to transfer from language to CS (L-CS transfer) and is justified in a way similar to what occurs in second language acquisition (human language to another language transfer).

Finding about the relationship above doesn't lead immediately to the impact of language on learning CS in general. This has motivated this work and can motivate more future works as well. Nevertheless, a non-experimental ex-post facto statistical analysis of CS students' records revealed that overall CS learning is enhanced in students of high English level proficiency.

References

Al-Nasser, A. S. (2015). Problems of English language acquisition in Saudi Arabia: An exploratory-cum-remedial study. *Theory and Practice in Language Studies, 5*(8), 1612-1619.

Aldmour, I., & Nylen, A. (2014, April, 11-13, 2014). *Impact of Cultural and Language Background on Learning Computer Science Concepts*. Paper presented at the International Conference on Teaching and Learning in Computing and Engineering (LaTiCE), 2014.

Burgstahler, S. (2011). Universal Design: Implications for Computing Education. *Trans. Comput. Educ., 11*(3), 1-17.

Chamot, A. U. (2004). Issues in language learning strategy research and teaching. *Electronic journal of foreign language teaching, 1*(1), 14-26.

Clancy, M. (2004). Misconceptions and attitudes that interfere with learning to program. In S. Fincher & M. Petre (Eds.), *Computer science education research* (pp. 85-100).

Ebrahimi, A. (1994). Novice programmer errors: Language constructs and plan composition. *International Journal of Human-Computer Studies, 41*(4), 457-480.

Goldman, K., Gross, P., Heeren, C., Herman, G. L., Kaczmarczyk, L., Loui, M. C., & Zilles, C. (2010). Setting the scope of concept inventories for introductory computing subjects. *ACM Transactions on Computing Education (TOCE), 10*(2), 1-29.

Jackson, J., Cobb, M., & Carver, C. (2005). *Identifying top Java errors for novice programmers.* Paper presented at the Proceedings Frontiers in Education 35th Annual Conference.

Lantolf, J. P., Thorne, S. L., & Poehner, M. E. (2015). Sociocultural theory and second language development. In B. VanPatten & J. Williams (Eds.), *Theories in second language acquisition: An introduction* (pp. 207-226).

Lee, O. (2005). Science education with English language learners: Synthesis and research agenda. *Review of Educational Research, 75*(4), 491-530.

Martirosyan, N. M., Hwang, E., & Wanjohi, R. (2015). Impact of English Proficiency on Academic Performance of International Students. *Journal of International Students, 5*(1), 60-71.

Miller, C. S. (2014). Metonymy and reference-point errors in novice programming. *Computer Science Education, 24*(2-3), 123-152.

Miller, C. S. (2016). *Human language and its role in reference-point errors.* Paper presented at the PPIG.

Nation, P. (2003). The role of the first language in foreign language learning. *Asian EFL journal, 5*(2), 1-8.

Nikolov, M., & Djigunović, J. M. (2006). Recent research on age, second language acquisition, and early foreign language learning. *Annual review of applied linguistics, 26*, 234-260.

Raheem, K. J. (2018). The Role of First Language on the Second Language Acquisition. *International Journal of Kurdish Studies, 4*(1), 110-117.

Ringbom, H., & Jarvis, S. (2009). The importance of cross-linguistic similarity in foreign language learning. In M. H. Long & C. J. Doughty (Eds.), *The handbook of language teaching* (pp. 106-118).

Tenenberg, J., & Knobelsdorf, M. (2014). Out of our minds: a review of sociocultural cognition theory. *Computer Science Education, 24*(1), 1-24.

Tobin, K., & McRobbie, C. J. (1996). Significance of limited English proficiency and cultural capital to the performance in science of Chinese-Australians. *Journal of Research in Science Teaching, 33*(3), 265-282.

Zendler, A., & Spannagel, C. (2008). Empirical foundation of central concepts for computer science education. *Journal on Educational Resources in Computing (JERIC), 8*(2), 1-15.

Zendler, A., Spannagel, C., & Klaudt, D. (2011). Marrying content and process in computer science education. *IEEE Transactions on Education, 54*(3), 387-397.

Zhai, C. (2012). *On the factors influencing language transfer in sla based on cognitive science.* Paper presented at the 2012 International Conference on Computer Science and Electronics Engineering.

Contact email: ismatdmour@gmail.com