

UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Implicit Learning and Computer Programming

Permalink

<https://escholarship.org/uc/item/26z3d703>

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 27(27)

ISSN

1069-7977

Authors

Mariotti, Francesco
Prevete, Roberto
Santoro, Matteo

Publication Date

2005

Peer reviewed

Implicit Learning and Computer Programming

Rebecca Mancy (mancyr@dcs.gla.ac.uk)

Centre for Science Education, Faculty of Education, University of Glasgow
St. Andrew's Building, 11 Eldon Street, Glasgow, G3 6NH, UK

Implicit Learning and Education

Implicit learning of regularities, using artificial grammars and similar protocols, has been a subject of investigation in psychology for half a century. It is therefore surprising that research concentrating on the role of implicit learning in the context of formal education is only just beginning to emerge. Although in the past several studies that have attempted in vain to uncover correlations between IQ and implicit learning, some (e.g. Buchner & Wippich, 2000) have suggested that this failure may simply reflect problems of test reliability. Indeed, Fletcher, Maybery and Bennett (2000) found significant positive correlations between IQ and implicit learning in a school setting. Also of relevance are studies by Pacton, Perruchet, Fayol and Cleeremans (2001) looking at the gradual integration of spelling patterns during childhood and adolescence.

I believe that programming is a fruitful area for further investigation of the role of implicit learning in formal education. Firstly, integrating structures and regularities is highly important in learning to program. In addition, competence can be differentiated from developmental stage, and whilst most (cognitively normal) people have no experience with programming, those who do may have learnt in a formal educational setting, or may be self-taught.

Rules and Regularities in Programming

The grammar of natural language is defined in a distributed way by its speakers and so is statistical: any rules we might define are purely descriptive and subject to exceptions. Programming language syntax on the other hand is determined in advance by strict rules programmed into the compiler or system and is therefore relatively easy to formulate as explicit rules. However, there are regularities in programming that are very hard to define explicitly, such as how and when to re-use code, or what makes a program elegant or readable.

Programming Education

In programming education, syntax and higher-level conceptual ideas are sometimes taught explicitly but often they are not. In some cases, this may be because the teacher omits to mention them; in others, it is because they are very difficult to define, and indeed the teacher him/herself is not aware of them or feels that they would be unhelpful to students. In these cases, students are forced to learn the patterns inductively, deriving the general principles from particular instances that they encounter. In some cases,

students may actively analyze programs in search of patterns, but they may also be learnt implicitly.

Interviews

I carried out a series of interviews with students on an introductory programming course at the University of Glasgow, asking learners to write a specific programming construct – a `while` loop – and to propose their own condition to determine when the loop would operate. To be legal, the condition must be a Boolean expression (i.e. a piece of code that evaluates to `True` or `False`) but whilst all students gave me correct Boolean expressions, many were unaware of this constraint on conditions. On questioning, one student who had previously not considered this relationship realized that the condition must always evaluate to either `True` or `False`, but many were unable to make this step.

It therefore appears that students can and do learn regularities in programming inductively, often without being (fully) aware of the rules that govern them.

Further work

I am currently investigating new ways to investigate the types of patterns that students can learn implicitly. One idea is to carry out further interviews with programmers of different levels of competence, to see whether regularities are picked up gradually over time. This would allow me to compare programmers who have received formal education and those who are self-taught, the latter group being perhaps more likely to have learnt rules implicitly.

Another possibility is to identify a pattern in some aspect of programming that is unlikely to have been actively considered by learners (e.g. in commenting) and investigate whether there is a relationship between integration of this regularity and programming competence in general.

References

- Buchner, A. & Wippich, W. (2000). On the Reliability of Implicit and Explicit Memory Measures. *Cognitive Psychology*, 40, 227-259.
- Fletcher, J., Maybery M. T., & Bennett, S. (2000) Implicit Learning Differences : A Question of Developmental Level? *Journal of Experimental Psychology: Learning, Memory and Cognition*, 26(1), 246-252.
- Pacton S., Perruchet P., Fayol M. & Cleeremans A. (2001) Implicit learning out of the lab: The case of orthographic regularities. *Journal of Experimental Psychology: General*, 130, 401-426.