



UNIVERSITÀ
DEGLI STUDI
DI BERGAMO | Dipartimento
di Scienze Umane
e Sociali



CQIA

Centro per la Qualità dell'Insegnamento,
dell'Innovazione Didattica e dell'Apprendimento
UNIVERSITÀ DEGLI STUDI
DI BERGAMO



BOOK OF PROCEEDINGS

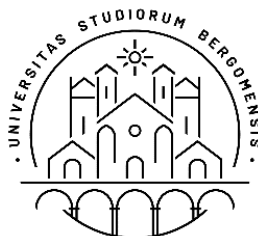
ATEE Spring Conference 2024

Teacher education research in Europe:
trends, challenges, practices and perspectives

May 29th – June 1st, 2024

S. Agostino, 2 - Bergamo, Italy

Edited by Nicole Bianquin and Francesco Magni



Università degli studi di Bergamo

2025

BOOK OF PROCEEDINGS ATEE Spring Conference 2024. Teacher education research in Europe: trends, challenges, practices and perspectives / Nicole Bianquin, Francesco Magni (edited by) - Bergamo: Università degli studi di Bergamo, 2025

ISBN: **978-88-97253-27-3**

DOI: [10.62336/unibg.978-88-97253-27-3](https://doi.org/10.62336/unibg.978-88-97253-27-3)

This publication is released under the Creative Commons
[Attribution Non-Commercial No Derivatives license \(CC BY-NC-ND 4.0\)](https://creativecommons.org/licenses/by-nc-nd/4.0/)



© 2025 The Authors

<https://aisberg.unibg.it/handle/10446/309209>

An event organised by:

Dipartimento di Scienze Umane e Sociali, **University of Bergamo** // www.dsus.unibg.it

CQIIA – Centro per la Qualità dell’Insegnamento, dell’Innovazione didattica e dell’Apprendimento, **University of Bergamo** // www.cqia.unibg.it

ATEE – Association for Teacher Education in Europe // www.atee.education

In collaboration with:

Siped
Società Italiana di Pedagogia
fondata nel 1989

SIPED
Società Italiana di
Pedagogia

BAUHAUS4EU
European University Alliance

Bahuhus4EU
European University
Alliance



CIRSE
Centro italiano per la
ricerca storico
educativa

SIRD
Società Italiana di Ricerca Didattica

SIRD
Società Italiana di Ricerca
Didattica

SIPSE
SOCIETÀ ITALIANA
PER LO STUDIO DEL
PATRIMONIO
STORICO-EDUCATIVO

SIPSE
Società Italiana per lo
Studio del Patrimonio
Storico-Educativo

**COMENIUS
ASSOCIATION
COMENIUS**

Comenius Association
European Higher Education
Network of Teacher and
Social Education

sipeges
società italiana di pedagogia generale e sociale

SIPeGeS
Società Italiana di
Pedagogia Generale e
Sociale

EFVET
European Forum
of Technical
and Vocational
Education and Training

EfVET
European forum of
Technical and Vocational
Education and Training

SIPeS
Società Italiana
di Pedagogia
speciale

SIPeS
Società Italiana di
Pedagogia Speciale

ETF
Working together
Learning for life
European Training Foundation

ETF
European Training
Foundation

**Centro
Italiano di
Ricerca
Pedagogica**

C.I.R.PED
Centro Italiano di
Ricerca Pedagogica

INVALSI

INVALSI
Istituto nazionale per
la valutazione del
sistema educativo di
istruzione e di
formazione

SIREF
Società Italiana di Ricerca Educativa e Formativa

SIREF
Società Italiana di
Ricerca Educativa e
Formativa

Ministero dell'Istruzione - Ufficio Scolastico Regionale e per la Lombardia
**Ufficio
Scolastico
Territoriale
di Bergamo**

**Ufficio Scolastico
Territoriale di Bergamo**

Conference Chair

Nicole Bianquin, associate professor in Special Education, University of the Aosta Valley, ATEE member;

Francesco Magni, associate professor in Education, University of Bergamo, ATEE member.

Scientific committee

Adolfo Scotto Di Luzio, Vice-Rector for Didactics, Guidance and Placement, University of Bergamo, Italy

Marco Lazzari, Head of the Department of Human and Social Sciences, University of Bergamo, Italy (until September 2024).

Anna Maria Falzoni, Director of CQIIA - Centre for Teaching Quality, Teaching Innovation and Learning, University of Bergamo, Italy

Maria Assunção Flores, University of Minho, Portugal

Joanne Banks, Trinity College of Dublin, Ireland

Federica Baroni, University of Bergamo, Italy

Tore Bernt Sorensen, University of Glasgow,

Scotland (UK)

Paolo Bertuletti, University of Bergamo, Italy

Serenella Besio, University of Bergamo, Italy

Antonio Borgogni, University of Bergamo, Italy

T.J. Ó Ceallaigh, University College Cork, Ireland;

ATEE AC member

Monica Crotti, University of Bergamo, Italy

Linda Daniela, University of Latvia, Latvia

Dietmar Frommberger, University of Osnabrück,

Germany

Paola Gandolfi, University of Bergamo, Italy

Mabel Giraldo, University of Bergamo, Italy

Etti Gordon Ginzburg, Oranim College of Education,

Israel

Michiel Heijnen, Marnix Academy, Netherlands; ATEE

AC President

Erika Kopp, Eötvös Loránd University, Hungary;

ATEE AC member

Marta Kowalczyk-Walędziak, University of Białystok,

Poland

Anna Lazzarini, University of Bergamo, Italy

Hagen Lehmann, University of Bergamo, Italy

Nicola Lovecchio, University of Bergamo, Italy

Gale MacLeod, University of Edinburgh, Scotland

(UK)

Fernando Marhuenda Fluixá, Universitat de València, Spain

Alessandra Mazzini, University of Bergamo, Italy

Sara Nijs, Leuven University, Belgium,

Katrin Poom-Valickis, University of Tallinn, Estonia

Andrea Potestio, University of Bergamo, Italy

Evelina Scaglia, University of Bergamo, Italy

Johannes Karl Schmees, Norwegian University of

Science and Technology (NTNU), Norway

Leah Shagrir, Levinsky College of Education, Israel;

ATEE AC member

Olena Shyyann, Lviv State University of Physical

Culture, Ukraine; ATEE AC member

Vasileios Symeonidis, Pädagogische Hochschule

Freiburg, Germany

Ronny Smet, Karel de Grote University of Applied

Science and Arts, Belgium; RDCs Coordinator of

ATEE

Agnieszka Szplit, Jan Kochanowski University of

Kielce, Poland; ATEE AC Vice-President

Elena Theodoropoulou, University of the Aegean,

Greece

Philippe Tremblay, Université de Laval, Québec

Vidmantas Tūtlys, Academy of Education, Vytautas

Magnus University, Lithuania

Wieland Wermke, Stockholm University, Sweden

Mara Westling Allodi, Stockholm University, Sweden

Jenny Wilder, Stockholm University, Sweden

Rano Zakirova Engstrand, Stockholm University,

Sweden

Organizing committee

Virginia Capriotti, University of Bergamo

Sara Cecchetti, University of Bergamo

Federico Chiappetta, University of Bergamo

Emilio Conte, University of Bergamo

Ester Guerini, University of Bergamo

Paolo Lazzaroni, University of Bergamo

Alice Locatelli, University of Bergamo

Isabel Maggiarra, University of Bergamo

Fabio Sacchi, University of Bergamo

Arianna Taravella, University of Bergamo

Programming errors and the attribution of intentionality to educational robots

Silvia Larghi, *University of Milano-Bicocca*, s.larghi1@campus.unimib.it
Edoardo Datteri, *University of Milano-Bicocca*, edoardo.datteri@unimib.it

Abstract

This paper explores, from a philosophical perspective, the connection between students' mental models of robots and their programming errors in educational robotics (ER). Pea (1986) identified the "superbug", a type of programming errors flowing from a misguided attribution of intentionality to computers. We substantiate, and illustrate with examples, the claim that a connection exists between students' mental models of robots and their programming errors, exploring the underlying assumptions. We then refine Pea's thesis arguing that the superbug does not result from the attribution of mental states to the system 'per se', but rather from the attribution of 'incorrect' ones. These reflections suggest possible connections between research on the attribution of mental states to robots, ER and computational thinking and provide insights for the design of teacher training.

Keywords: mental state attribution to robots; psychology of computer programming; educational robotics; mental models; programming errors.

1. Introduction

The use of robots as mediators of learning in education is well established (Bano et al., 2024; Uslu et al., 2022; Anwar et al., 2019; Benitti, 2012). They can be used to facilitate teaching and learning in STEAM – Science, Technology, Engineering, Art, and Mathematics (Sapounidis et al., 2024; González et al., 2020), to stimulate scientific thinking (Datteri & Zecca, 2016; 2017), to support the acquisition of a second language (Mubin et al., 2013). They can be used to foster the acquisition of cross-disciplinary social-relational and emotional skills and to develop and enhance cognitive skills, such as those related to problem solving and computational thinking (Ching & Hsu, 2024; Zhang et al., 2021). Typically, educational robotics (ER) activities involve the use of robots as programming platforms. In this context, students are invited to program robots so that they display certain behaviors or exhibit certain capacities. To effectively guide these activities, assess their outcomes and understand their dynamics, it is essential to analyze the nature and causes of the programming errors made by students.

Building on the tradition of the psychology of computer programming (Weinberg, 1971; Sheil, 1981), this paper presents a philosophical reflection on the relationship between students' programming errors and their understanding of robots. Following the insight that computer programming activities should be studied from a psychological perspective, programming errors have been conceptualized in literature as the result of "faulty mental models" – to this respect see, for example, the literature review by McCauley and colleagues (2008) on learning and teaching debugging – a perspective that has been revisited and developed in more recent work, such as that of O'Dell (2017).

In this framework, we start from the consideration that programming errors may be caused by students' conceptions, or mental models, of robots and computer systems: 'poor' mental models of a system can lead to errors that result in system malfunctions. Pea (1986) introduced the concept of "superbug", a category of programming errors arising from a misguided attribution of intentionality to computers. This attribution causes programmers to assume that a machine can go "beyond the information given" in the code. He identified three forms of superbug that can be often observed among novice programmers: the "parallelism bug", the "intentionality bug" and the "egocentrism bug", and argued that all these three forms of superbug arise because programmers unconsciously assume the presence of "a hidden mind" within the programming language, endowing the computer with intelligent interpretive capabilities it does not possess (Pea, 1986).

This paper has two goals. The first one is to substantiate, and illustrate with examples, the claim that a connection exists between students' mental models of robots and their programming errors, consistent with Pea's thesis. In doing so, we introduce and discuss three assumptions possibly underlying the superbug: the global view assumption, the rationality assumption, and the mind-reading assumption. The second goal is to challenge Pea's claim that the attribution of intentionality to robots necessarily causes the superbug. Instead, we propose that the superbug arises from the attribution of *incorrect* mental states and capacities to computers, not from the attribution 'per se'.

Addressing these goals, this paper aims to suggest that Pea's insight can shed light on the cause of programming errors and foster a connection between the emerging research on the attribution of mental states to artificial systems (Thellman et al., 2022) and the research on computational thinking (CT) (Denning & Tedre, 2019) and ER. More specifically, the reflection proposed aims to contribute to the development of theoretical frameworks that can inform the empirical study of programming errors made by novice programmers in CT and ER activities. A deeper understanding of the relationship between individuals' conceptualization of robots and AI technologies and their programming errors may offer valuable insights into how this dimension can be integrated in teaching and learning models and how teacher education programs should be designed to address it.

The paper is structured as follows. In Section 2 we discuss how people model robots, focusing on mentalistic modeling and the attribution of mental states to the system. In Section 3 we address the first goal, discussing the relationship between the attribution of mind to robots and programming errors, and examining three assumptions that may be underlying the "superbug". In Section 4 we address the second goal, arguing, in accordance with Dennett (1971, 1987) that the 'hidden mind assumption' is not necessarily an obstacle to learning how to program. In Section 5 we draw some conclusions, provide considerations on the possible connection between research on mental state

attribution to robots and research on ER and CT, and outline potential implications for the design of teacher training programs.

2. The attribution of mind to robots

Research in human-robot interaction suggests that individuals may adopt different modeling strategies when interacting with robots, ranging from taking non-mentalistic to mentalistic stances. Taking a non-mentalistic stance towards a robot implies the adoption of an explanatory and predictive strategy which does not refer to the system's mind, often relying on the theoretical vocabularies of physics or electronics (e.g., "the system is malfunctioning because the battery is low"). In contrast, taking a mentalistic stance entails the attribution of mental states and mental capacities to the system.

Thellman and colleagues (2022) provide a review on the attribution of mental states to robots exploring how the phenomenon has been investigated through various angles. One of these is based on the concept of Theory of Mind (ToM) (for a general discussion, see Carruthers & Smith, 1996; Premack & Woodruff, 1978). The possession of a ToM implies the ability to represent oneself and others as entities with mental states, characterized by Griffin and Baron-Cohen (2002) as intentional, content-bearing and representational. Mental states, in this perspective, are typically expressed in the form of propositional attitudes, such as "believing that p" or "knowing that q". Several studies have investigated whether humans develop a ToM towards robots and other artificial agents (e.g., Banks, 2020; Zhang, 2019).

Another fundamental perspective orienting empirical research on the attribution of mental states to robots is Dennett's intentional systems theory (1971; 1987). Dennett identifies three possible stances that individuals can adopt to explain and predict the behavior of a system: the physical stance, where explanations and predictions are based on the laws of physics, the design stance, which refers to the system's design and the intentional stance. Taking the intentional stance towards a system consists in adopting an explanatory and predictive strategy that attributes to the system beliefs, desires, and other propositional attitudes, assuming it will act in the most rational way given its current beliefs and desires. A wide number of studies have explored whether people adopt an intentional stance towards robots of various types, and under what conditions this phenomenon occurs (Perez-Osorio & Wykowska, 2020).

Empirical research on the attribution of mental states to robots shows that the phenomenon can be influenced by both human factors and robot related factors. Among human factors there are age, motivation, cultural and socioeconomic background, interaction history; robot related factors are typically related to the robot's behavior and appearance (Thellman et al., 2022).

3. Mind attribution to robots and programming errors

3.1 The psychology of computer programming

Gerald M. Weinberg's book "The Psychology of Computer Programming," first published in 1971, explored the psychological, social and organizational aspects of programming by analyzing not only the technical processes involved in writing code, but also the interpersonal and cognitive dynamics that influence the work of programmers. The message was that the very activity of computer programming should be studied from a psychological perspective, envisioning "computer programming as a human activity" as a new field of study (Weinberg, 1971). Sheil's review on "The Psychological Study of Programming" (1981), summarized and evaluated the psychological research on programming available up to that time, offering a basis for the study of programming from a psychological perspective.

More recent studies approach the topic from various perspectives, through a variety of methods. For example, Lin and colleagues (2015) study students' cognitive processes during program debugging using an eye tracker. Storjak and colleagues (2022) explore the mental models that primary school

children develop in relation to robots and programming. According to O'Dell (2017), programmers rely on approximations of the system's behavior (i.e., mental models of the system) to guide the development. Mental models help programmers in reasoning about the system's behavior, however, as approximations, they can sometimes be incorrect, leading to software failures.

A central idea, in the framework of the psychology of computer programming, is that mental models formed by programmers about the system, as well as about the programming language - for example the attribution to the system of beliefs, desires, intentions, in case a mentalistic modeling strategy is adopted - may influence the cognitive mechanisms triggered during the programming process. The resulting program may be impacted by the assumptions on the system made by the programmer and 'wrong' mental models may produce programming errors. Understanding programmers' mental model is therefore essential to make sense of their programming errors.

3.2 The superbug and the underlying assumptions

In ER activities, programming errors may be caused by students' conceptions, or mental models of the robot or of the complex system represented by the syntax and semantics of the programming language. Students, programming robots, may adopt either mentalistic or non-mentalistic strategies to model the system. In the following, we explore the idea that the adoption of a mentalistic modeling strategy may lead to programming errors in novice programmers. This idea has been initially voiced by Pea (1986). Pea (1986) identified the "superbug": a type of programming error flowing from a misguided attribution of intentionality to computers.

«The 'superbug' may be described as the idea that there is a hidden mind somewhere in the programming language that has intelligent, interpretive powers. It knows what has happened or will happen in lines of the program other than the line being executed; it can benevolently go beyond the information given to help the student achieve her goals in writing the program. This "hidden mind superbug" interpretation provides a deep explanation of the various misconceptions that plague the novice programmer. » (Pea, 1986, pp. 32-33).

In line with Pea's thesis, we present some examples (derived from real student programs) that suggest the existence of a connection between students' mentalistic modeling of robots or programming languages and the programming errors they make. Programmers' mental models of a system include a set of assumptions about the system, reference to which can be used to explain and predict its behavior. Some of these assumptions may underlie the "superbug" (i.e. the idea that there is a hidden mind in the system). Consider a Braitenberg vehicle (Braitenberg, 1986) equipped with two lateral light sensors (Figure 1: sensor 1 is mounted on the left, and sensor 2 on the right) and programmable. Students are tasked with the following request: "Program the robot to exhibit the following behavior: the robot must move forward, avoiding the lights."

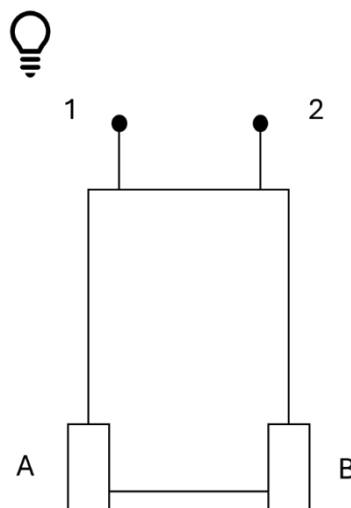


Figure 1: Braitenberg vehicle equipped with two lateral light sensors.

```

FOREVER {
  IF (left-sensor == light) THEN turn right
  IF (right-sensor == light) THEN turn left
  IF ((left-sensor == light) and (right-sensor == light)) THEN
    go backwards
  go forward
}

```

Figure 2: A student provides the following program (pseudocode).

This implementation presents two major issues. First, it results in an unconditional forward movement: the robot moves forward regardless of whether it detects light or not. Second, there are sequential execution limitations: the sequence of <<if - then>> conditions may not function as intended in dynamically changing environments. The first instruction executed is the test of the <light on the left> condition. If, for example, a light source is near the right sensor (Figure 2), the first <<if>> condition might fail to detect it (resulting in no avoidance). The <light on the right> condition is tested when the second <<if>> condition is executed, sequentially after the first one. However, when the second <<if>> condition is executed, the light might, in the meantime, have shifted position (Figure 3), causing the left sensor to fail to detect it as well, again resulting in no avoidance.

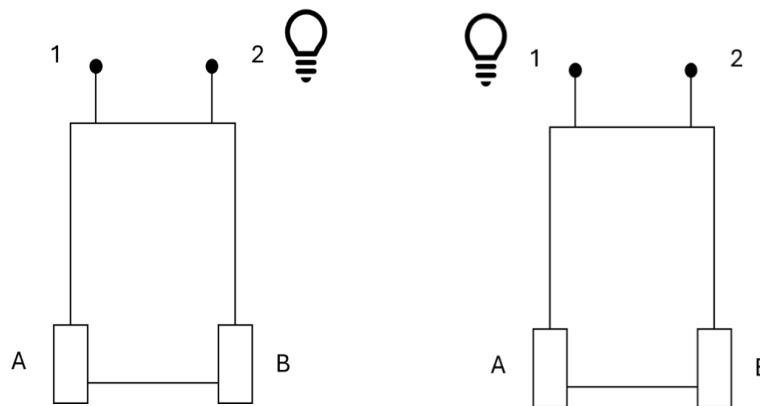


Figure 3 and 4

At the basis of this programming strategy there might be the idea that all instructions are 'considered' simultaneously, without taking into account that they are executed sequentially according to how the program is written, what we call the global view assumption. The programmer attributes to the system (or to the programming language) the ability to 'choose' the 'right' instruction to execute, assuming its knowledge of its internal state and external conditions. What would the 'right' instruction be? Linked to the global view assumption there is the rationality assumption: the programmer assumes that the robot (or the programming language) is able to autonomously select the most rational <<if - then>> structure based on the given circumstances and its implicit goals. The programmer attributes a goal to the system (not explicitly defined in the code) that guides its choice of the most appropriate action. Attributing rationality to the system is tantamount to taking the intentional stance. In Dennett's framework (Dennett, 1971, 1987), adopting the intentional stance towards a system consists in adopting an explanatory and predictive strategy that attributes beliefs, desires, other intentional states to it, and assuming that the system will consistently act in the most rational way based on its current beliefs and desires. To address the above issues, the student proposes the following program.

```

FOREVER {
    IF (left-sensor == light) THEN turn right; OTHERWISE
        IF (right-sensor == light) THEN turn left; OTHERWISE
            IF ((left-sensor == light) and
                (right-sensor == light)) THEN go backwards
    }

```

Figure 5: the student proposes the following program to address the above issues.

This revised implementation introduces what might be called the mind-reading assumption. Here, the system lacks an explicit “move forward” command. Instead, the student assumes that the robot (or the programming language) has an implicit understanding of the goal of the programmer - avoiding light - and acts accordingly, beyond the explicit instructions provided. The programmer attributes to the system the possession of a theory of the programmer’s mind (a second order mental state), where the programmer ascribes to the system the capability to think to what the programmer themselves is thinking (i.e., what is their goal in developing the program).

The examples discussed above¹ aim to illustrate that a possible connection between students’ mentalistic modeling of robots or programming languages and their programming errors may exist. The global view assumption, the rationality assumption, and the mind-reading assumption offer potential explanations for the observed programming choices; however, further systematic studies, such as those employing think-aloud protocols, student interviews, or analyses of debugging practices, would be necessary to determine whether these programming decisions are genuinely influenced by the assumptions discussed or arise from other factors.

4. Robots can be usefully modeled as mental agents

We try now to refine Pea’s thesis by developing the idea that the superbug is not the attribution of mental states and capacities to the robot or to the programming language per se, but rather the attribution of the wrong mental states and capacities to it.

On the one hand, as we discussed in the previous section, the attribution of intentionality and rationality to robots and programming languages could result in systematic programming errors (superbugs). On the other hand, however, the tendency to mentalize may prove to be a useful strategy, especially in the early stages of programming learning. Attributing goals, intentions or beliefs to the robot can help students to effectively predict the behavior of the system. As Dennett (1971, 1987) points out, the intentional stance can be predictively useful, and in some circumstances offers significant advantages over other ways of conceptualizing the system. According to Dennett, the tactic of adopting the intentional stance is likely to be effective when there is reason to believe that the assumption of optimal design is justified, and when predicting behavior from the design or physical stance is impractical. The intentional stance (which involves treating a system as an ‘intentional agent’, assuming that it possesses beliefs, desires, and intentions) works because it provides a pragmatic and effective approach to explaining and predicting the behavior of complex entities, whether they are humans, animals, or artificial systems (Dennett, 1971; 1987).

¹ It is possible to identify a link between the three assumptions described, the *global view assumption*, the *rationality assumption* and the *mind-reading assumption*, with the three types of ‘superbug’ identified by Pea (1986), respectively the “parallelism bug”, i.e. the mistaken belief that sequentially ordered lines of code can execute simultaneously, the “intentionality bug” which consists in attributing the ability to go “beyond the information given” to the computer or program itself, taking an intentional stance towards the system (Dennett, 1971), and the “egocentrism bug”, based on the assumption that the programming language is able to understand the programmer’s goals without those goals being explicitly represented in the code.

The key to avoiding the programming errors that Pea refers to is for the programmer to attribute to the system goals (beliefs, desires, intentions) that the system actually has; the superbug arises when the mental model of the system's mind is, in some sense to be clarified, wrong'.

For example, suppose the programmer is faced with a Python function that implements a bubblesort algorithm designed to operate on arrays of integers. There is a clear sense in which the programmer can bypass the computational language and attribute to the function the goal of ordering integers. If the programmer calls this function on an array of integers, no programming error is made. Now suppose the programmer mistakenly assumes that the program has a theory of their mind and is therefore able to 'understand', beyond the information given, what types of values the programmer wants to order. They will call the Python function on, say, an array of characters, causing an execution error. Following Pea's intuition, we could argue that, in novice programmers, this decision could be caused by misattributing mental and interpretive capabilities to the program. The point here is that this attribution would not be mistaken because it attributes mental capacities to the system, but because it attributes to it the 'wrong' mental capacities. There is nothing wrong in reconstructing a program in mentalistic terms, provided that the reconstruction is 'right' in some sense of the term, for example it is explanatory, predictively adequate and mirroring the algorithmic structure of the program.

5. Conclusions

We argued that a connection exists between students' mental models of robots (and programming languages) and their programming errors, in line with Pea's thesis, and explored the assumptions that may underlie the emergence of the superbug. Moreover, we argued that attributing intentionality to robots need not necessarily cause the superbug. Instead, the superbug arises from the attribution of incorrect mental states and capabilities to the system. The key to avoiding the superbug is for programmers to attribute to the system goals (beliefs, desires, intentions) that the system actually possesses. To sum up, mentalizing can be useful, provided that mental models are accurate.

The discussion presented in this paper highlights a possible intersection of two distinct, yet interrelated research domains: the research on the attribution of mental states to robots and the study of ER and CT. On the one hand, insights from research on ER and CT could enhance the understanding of mental state attribution to robots addressing critical questions such as: what cognitive abilities do programmers attribute to computers? How are these attributions influenced by their background and prior experience? On the other hand, research on the attribution of mental states to robots may offer new perspectives on learning and teaching processes in ER and CT. For example, it can inform the research on learning and teaching of programming and debugging and provide insights on how teacher education programs should be designed to integrate the dimension of mental state attribution to machines.

Understanding how novice programmers conceptualize robots (and programming languages), and whether and how these conceptualizations could lead to systematic programming errors, has direct implications for teacher education. If teachers and educators are to effectively support students in ER and CT activities, they need to be able to recognize the cognitive assumptions and mental models that may underlie students' programming errors. This implies developing diagnostic pedagogical skills that allow educators to identify whether a student's programming error may stem, for instance, from a 'global view' or 'mind-reading' or 'rationality' assumption, or, more broadly, from a misguided attribution of intentionality to the robot (or the programming language). Teacher training programs should therefore include training that familiarizes teachers and educators with possible mentalistic modeling strategies of robots (and programming languages). Moreover, integrating knowledge and reflections on the mentalization of machines into teacher education may help teachers and educators to understand how and when the attribution of mental states to machines can be either an effective strategy in ER and CT activities or become misleading.

In conclusion, the reflections proposed suggest that the integration of perspectives from research in human-robot interaction could foster valuable insights for contemporary studies on ER and CT.

Bibliography

- Anwar, S., Bascou, N. A., Menekse, M., & Kardgar, A. (2019). A systematic review of studies on educational robotics. *Journal of Pre-College Engineering Education Research (J-PEER)*, 9(2), 2.
- Bano, S., Atif, K., & Mehdi, S. A. (2024). Systematic review: Potential effectiveness of educational robotics for 21st century skills development in young learners. *Education and information technologies*, 29(9), 11135-11153.
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978-988.
- Banks, J. (2020). Theory of mind in social robots: replication of five established human tests. *International Journal of Social Robotics*, 12(2), 403-414.
- Braitenberg V. (1986) *Vehicles: Experiments in synthetic psychology*. MIT press
- Carruthers, P. & Smith, P. K. (1996). *Theories of Theories of Mind*. Cambridge University Press.
- Ching, Y. H., & Hsu, Y. C. (2024). Educational robotics for developing computational thinking in young learners: A systematic review. *TechTrends*, 68(3), 423-434.
- Datteri, E., & Zecca, L. (2016). The game of science: an experiment in synthetic roboethology with primary school children. *IEEE Robotics & Automation Magazine*, 23(2), 24-29.
- Datteri, E., & Zecca, L. (2017). Theoretical vocabularies and styles of explanation of robot behaviours in children. *Research on Education and Media*, 9(1), 3-9.
- Dennett, D. C. (1971). Intentional systems. *The Journal of Philosophy*, 68(4), 87-106.
- Dennett, D.C. (1987). *The Intentional Stance*.
- Denning, P. J., & Tedre, M. (2019). *Computational thinking*. Mit Press.
- González, M.Á., Rodríguez-Sedano, F.J., Llamas, C.F., Gonçalves, J., Lima, J., & García-Peñalvo, F.J. (2020). Fostering STEAM through challenge-based learning, robotics, and physical devices: A systematic mapping literature review. *Computer Applications in Engineering Education*, 29, 46 - 65.
- Griffin, R. & Baron-Cohen, S. (2002) *The intentional stance: Developmental and neurocognitive perspectives*. Daniel Dennett, 83-116. 2002.
- Lin, Y. T., Wu, C. C., Hou, T. Y., Lin, Y. C., Yang, F. Y., & Chang, C. H. (2015). Tracking students' cognitive processes during program debugging—An eye-movement approach. *IEEE transactions on education*, 59(3), 175-186.
- McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: a review of the literature from an educational perspective. *Computer Science Education*, 18(2), 67-92.
- Mubin, O., Shahid, S., & Bartneck, C. (2013). Robot assisted language learning through games: A comparison of two case studies.
- O'Dell, D.H. (2017). The debugging mind-set. *Communications of the ACM*, 60, 40 - 45.
- Pea, R. D. (1986). Language-Independent Conceptual "Bugs" in Novice Programming. *Journal of Educational Computing Research*, 2(1), 25-36.
- Perez-Osorio, J., & Wykowska, A. (2020). Adopting the intentional stance toward natural and artificial agents. *Philosophical Psychology*, 33(3), 369-395.
- Premack, D., & Woodruff, G. (1978). Does the chimpanzee have a theory of mind?. *Behavioral and brain sciences*, 1(4), 515-526.
- Sapounidis, T., Tselegkaridis, S., & Stamovlasis, D. (2024). Educational robotics and STEM in primary education: a review and a meta-analysis. *Journal of Research on Technology in Education*, 56(4), 462-476.
- Sheil, B. A. (1981). The psychological study of programming. *ACM Computing Surveys (CSUR)*, 13(1), 101-120.
- Storjak, I., Krzic, A. S., & Jagust, T. (2022). Elementary school pupils' mental models regarding robots and programming. *IEEE transactions on education*, 65(3), 297-308.
- Thellman, S., De Graaf, M., & Ziemke, T. (2022). Mental state attribution to robots: A systematic review of conceptions, methods, and findings. *ACM Transactions on Human-Robot Interaction (THRI)*, 11(4), 1-51.
- Uslu, N.A., Yavuz, G.Ö., & Usluel, Y.K. (2022). A systematic review study on educational robotics and robots. *Interactive Learning Environments*, 31, 5874 - 5898.
- Weinberg, G. M. (1971). *The psychology of computer programming* (Vol. 29). New York: Van Nostrand Reinhold.
- Zhang, Y., Song, W., Tan, Z., Wang, Y., Lam, C. M., Hoi, S. P., ... & Yi, L. (2019). Theory of robot mind: False belief attribution to social robots in children with and without autism. *Frontiers in psychology*, 10, 1732.
- Zhang, Y., Luo, R., Zhu, Y., & Yin, Y. (2021). Educational Robots Improve K-12 Students' Computational Thinking and STEM Attitudes: Systematic Review. *Journal of Educational Computing Research*, 59, 1450 - 1481.