
Towards an intelligent assessment system for evaluating the development of algorithmic thinking skills

An exploratory study in Swiss compulsory schools

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera Italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Giorgia Adorni

under the supervision of
Prof. Luca Maria Gambardella and Prof. Alberto Piatti

February 2025

Dissertation Committee

Prof. Cesare Alippi	Università della Svizzera Italiana, Switzerland
Prof. Monica Landoni	Università della Svizzera Italiana, Switzerland
Prof. Engin Bumbacher	Haute école pédagogique Vaud, Switzerland
Ph.D. Marc Lafuente Martínez	

Dissertation accepted on 7 February 2025

Research Advisor

Prof. Luca Maria Gambardella

Co-Advisor

Prof. Alberto Piatti

PhD Program Director

Prof. Walter Binder & Prof. Stefan Wolf

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Giorgia Adorni
Lugano, 7 February 2025

Preoccupied with a single leaf...
you won't see the tree.
Preoccupied with a single tree...
you'll miss the entire forest.
Don't be preoccupied with a single
spot.
See everything in it's entirety...
effortlessly.
That is what it means to truly "see".
Takuan Soho

Abstract

The rapid digitalisation of contemporary society has profoundly impacted various facets of our lives, including healthcare, communication, business, and education. The ability to engage with new technologies and solve problems has become crucial, making computational thinking (CT) skills, such as pattern recognition, decomposition, and algorithm design, essential competencies. In response, Switzerland has undertaken considerable research and initiatives aimed at integrating CT into the educational system, preparing students for the digital age.

This research aims to contribute to these efforts by developing a comprehensive framework for large-scale assessment of CT skills throughout the Swiss compulsory education system, with a particular focus on algorithmic thinking (AT), which pertains to the ability to design algorithms. To achieve this, we first developed a competence model that captures the situated and developmental nature of CT, enabling the design of activities tailored to varying cognitive abilities, learner age, and the contexts in which they occur. This framework not only clarifies how the characteristics and components of these activities influence the development of CT competencies but also provides guidance for effectively assessing them.

A key contribution of this research is the development of an assessment activity to measure AT skills on a large scale. The activity is designed in two variants: one uses non-digital artefacts (unplugged format) and provides a manual expert assessment, while the other relies on digital artefacts (virtual format), automating the assessment process. To provide a more comprehensive evaluation of students' competencies, we developed an Intelligent Assessment System based on Bayesian Networks with noisy gates, which offers real-time probabilistic assessment for each skill rather than a single overall score.

The results of this study indicate that the proposed instrument can measure AT competencies across different age groups and educational contexts in Switzerland, demonstrating its applicability for large-scale use. The findings suggest that AT competencies exhibit a progressive development, with no overall gender differences, though variations are observed at the school level. Several factors, including the type of artefact-based environment and the situated context, influenced AT performance sig-

nificantly. These results underscore the importance of creating assessment tools that are both accessible and adaptable to various contexts. Additionally, they highlight the need for careful and nuanced interpretation of the data, considering the diverse factors that may impact student performance and the validity of the assessments across different settings. In conclusion, this instrument holds significant potential for integration into real classroom settings, providing a scalable solution for assessing AT skills across a wide range of educational environments.

Acknowledgements

I want to express my profound thanks to my dissertation supervisors, **Luca Maria Gambardella** and **Alberto Piatti**, for their guidance, mentorship, and constructive feedback throughout the entire research process, which has helped me grow both academically and professionally.

My sincere gratitude goes to the members of my committee. **Engin Bumbacher**, who provided insightful academic advice and collaborated with me in discussing the definition of one of the theoretical frameworks I developed. **Monica Landoni**, who offered support and guidance on some of my works and included me in the TecLadies initiative. **Marc Lafuente Martinez**, though I had limited direct contact with him, suggested relevant literature that contributed to my research. **Cesare Alippi**, also one of the professors I assisted during my doctoral studies, provided valuable feedback from a unique engineering perspective, complementing the insights of others.

I would also like to thank my academic peers and colleagues who have supported me throughout my doctoral journey. The EPFL group, including **Francesco Mondada**, **Laila El-Hamamsy**, **Kunal Massé**, and **Jérôme Brender**, despite the geographical distance, has always been available for discussions, offering support, and advice that significantly contributed to the success of my work. The School of Education group at FHNW, including **Dorit Assaf** and **Elia Lutz**, for their contributions to the project and their assistance with data collection in the German-speaking cantons of Switzerland. The DFA group, including **Lucio Negrini**, who introduced me to ROTECO and supported the project, **Moro Thierry** for his assistance with the platform installation, setup and testing, and **Masiar Babazadeh** who supported me during the Swiss TecLadies initiative. The IDSIA group, including **Francesca Mangili** and **Alessandro Antonucci** for their support in understanding the theories behind the probabilistic models, **Claudio Bonesana** and **David Huber** for their help with the implementation of these models, as well as their support with pre-existing libraries and technical resources, **Giovanni Profeta** for providing valuable UX design feedback for my platform, and **Sandra Mitrovic** for involving me in Swiss TecLadies and in other collaborations that unfortunately have not materialised yet. **Igor Artico**, initially supported me on a personal level and later offered valuable academic advice and assistance with one of my recent papers. Finally,

the students I supervised for their Bachelor thesis project, including **Simone Piatti** and **Volodymyr Karpenko**, who contributed to the implementation of part of the platform are credited in the related paper, and **Samuel Corecco** with whom I co-authored another paper.

I would also like to acknowledge the *Swiss National Science Foundation* (SNSF), under the National Research Program 77 (NRP-77) Digital Transformation (No: 407740_187246), for financially supporting my work, and express my gratitude to all the *participants* in my research, whose time and valuable contributions were crucial to the success of this project.

On a more personal note, I would like to thank my family. My father, **Stefano**, my strongest pillar and my grandparents, **Ariodante** and **Giulia**, always a source of inspiration. My brother, **Ario**, and my mother, **Stefania**, contributing in their own ways. My aunt, **Lorenza**, always there for me like a second mother, and the rest of my family for their presence in my life. A special mention to **Kira**, my dog, whose unconditional love and companionship have been a constant source of joy.

A special thank you to my partner **Lorenzo**. Despite our relatively short time together, he has been by my side every step of the way, supporting me in countless ways. His kindness, sincerity, and sense of humour never fail to brighten my days. His love, patience, and unwavering belief in me have kept me going through the most challenging times.

A mention goes to my office mates who have shared my everyday work life: Hubi, Filip, Ele, Tuls, Koppány and Ambro; the people from the 4th floor: Alle, Luca, Taimoor, Alex; and the incredible friends I've made at IDSIA: Step, Franca, Matte, Nicho, Elia, Luca, Simo, Milad, Angelo, Dario, and Davide.

I would also like to thank those who, at different moments, have been part of this journey and enriched it: Ted, Lambe, Lollo, Nicola, Pietro, Marco, Claudio, Luca, Palla and all the members of the university choir.

A big thank you to the friends who, even from a distance, have been there for me throughout: Teo, Nassim, Lore Benatti, Lore Bini, Fede and Chri.

Finally, heartfelt thanks to the friends who have been with me from the very beginning: Beba and Manu.

Thank you all for your friendship, support, and love.

Contents

Contents	ix
Figures	xv
Tables	xix
Acronyms	xxii
I Introduction and Related works	1
1 Introduction	2
1.1 Research context	2
1.1.1 Switzerland's education system	2
1.2 Research motivation	3
1.3 Research objectives	4
1.3.1 Project overview	4
1.3.2 Doctoral research focus	5
1.4 Research questions	5
1.5 Research challenges	6
1.5.1 Diversity of the population	6
1.5.2 Sample size considerations	8
1.5.3 Impact of the COVID-19 pandemic	8
1.5.4 Ethical considerations	9
1.6 Research contribution	10
1.7 Structure of the thesis	11
2 Related works	12
2.1 Computational Thinking (CT) in education	12
2.2 Algorithmic Thinking (AT) in education	13
2.3 Defining CT	14

2.3.1	A situated cognition perspective	15
2.3.2	A developmental perspective	15
2.4	CT and AT integration in education	17
2.4.1	Global overview	17
2.4.2	European context	17
2.4.3	Swiss context	18
2.5	Assessing AT	19
2.6	Intelligent Assessment Systems (IASs)	22
2.6.1	Learner modelling and competence profiling	23
2.6.2	Probabilistic graphical models	23
2.7	Gaps in existing research and the contribution of this study	26
2.7.1	Defining a competence model for CT - RQ1	26
2.7.2	Developing assessment instruments for AT - RQ2	27
2.7.3	Designing a probabilistic IAS for AT assessment - RQ3	28
2.7.4	Understanding AT competencies in the Swiss educational context - RQ4	28
II	Methodological frameworks	29
3	A framework for the design and the assessment of CT activities	30
3.1	Summary	30
3.2	Definition of CT	31
3.3	The CT-cube	33
3.3.1	Framework applications	34
4	A framework for the analysis and design of CTPs	35
4.1	Summary	35
4.2	Definition of CTP	36
4.2.1	Components	36
4.2.2	Characteristics	38
4.3	Catalogue of CT competencies	41
4.4	Profiling CTPs	46
4.4.1	Framework applications	47
5	Probabilistic modelling for IASs	49
5.1	Summary	49
5.2	Bayesian Networks (BNs) and learner models	50
5.3	Noisy gates	52
5.3.1	Disjunctive gates	52

5.3.2	Leaky models	53
5.3.3	Comparison with Bayesian Knowledge Tracing (BKT)	54
5.4	Assessment rubrics	55
5.5	Modelling assessment rubrics by BNs	56
5.5.1	Ordering of competences	56
5.5.2	Supplementary competencies	58
III	Assessment instruments	60
6	The unplugged Cross Array Task (CAT)	61
6.1	Summary	61
6.2	The cross array	62
6.3	Activity design	63
6.3.1	CT-cube dimensions of the CAT	65
6.4	Activity profile	67
6.4.1	Components	67
6.4.2	Characteristics	68
6.4.3	Competencies	69
6.5	Competencies assessment	70
6.5.1	Algorithm dimension	71
6.5.2	Interaction dimension	73
6.5.3	CAT score	73
6.5.4	Task metrics	74
7	The virtual Cross Array Task (CAT)	75
7.1	Summary	75
7.2	Activity design	76
7.3	Activity profile	79
7.3.1	Components	79
7.3.2	Characteristics	80
7.3.3	Competencies	81
7.4	Competencies assessment	82
7.4.1	Algorithm dimension	83
7.4.2	Adjusted algorithm dimension	83
7.4.3	Interaction dimension	83
7.4.4	CAT score	84
7.4.5	Task metrics	84
7.5	Instrument development and implementation	85
7.5.1	Development process	85

7.5.2	Implementation	90
7.5.3	Prototypes	97
8	The IAS for the CAT	107
8.1	Summary	107
8.2	Modelling the CAT assessment rubric	108
8.2.1	Ordering of competencies	109
8.2.2	Answers encoding	110
8.2.3	Supplementary competencies	111
8.3	Parameters' elicitation	112
IV	Results	115
9	Experimental study on the unplugged CAT	116
9.1	Summary	116
9.2	Methodology	117
9.2.1	Study context	117
9.2.2	Participant selection	117
9.2.3	Data collection approach and procedures	118
9.2.4	Data analysis approach	118
9.3	Results	119
9.3.1	Participation and performance	119
9.3.2	Competencies development	123
9.3.3	Algorithms classification	125
10	Experimental study on the virtual CAT (pilot)	127
10.1	Summary	127
10.2	Methodology	128
10.2.1	Study context	128
10.2.2	Participant selection	128
10.2.3	Data collection approach and procedures	129
10.2.4	Data analysis approach	130
10.3	Results	130
10.3.1	Participation and performance	130
10.3.2	Competencies development	133
11	Experimental study on the virtual CAT (main)	134
11.1	Summary	134
11.2	Methodology	135

11.2.1 Study context	135
11.2.2 Participant selection	135
11.2.3 Data collection approach and procedures	136
11.2.4 Data analysis approach	136
11.3 Results	138
11.3.1 Participation and performance	138
11.3.2 Competencies development	142
12 Factors influencing CAT performance	147
12.1 Summary	147
12.2 Methodology	147
12.3 Results	148
12.3.1 Model selection and refinement	148
12.3.2 Model application and predictor significance	152
12.3.3 Domain-specific analysis	160
13 Competencies assessment with IAS	163
13.1 Summary	163
13.2 Methodology	164
13.3 Evaluation of the unplugged CAT data	164
13.4 Real-time evaluation of the virtual CAT data	170
V Discussion and Conclusion	176
14 Summary and interpretation of findings	177
14.1 Developing an age-based competence model for CT	177
14.2 Developing a large-scale assessment instrument for AT	178
14.3 eloping an IAS	179
14.4 Examining AT competencies in Swiss educational settings	180
15 Practical implications	183
16 Limitations and Future works	185
16.1 Extending and validating the competence models	185
16.2 Instrument potential for learning and teaching	186
16.3 Integrating tutoring capabilities	186
16.4 Evaluating predictive power of the IAS	188
16.5 Instrument extension and validation	188
16.6 Instrument integration	189

17 Conclusions	191
VI Appendices	192
A Participant information sheets and parental consent forms	193
B Linking CTP characteristics to CT competencies	198
B.1 Problem setting competencies	200
B.2 Algorithmic competencies	202
B.3 Assessment competencies	205
C Main study with the unplugged CAT	208
C.1 Protocol template for administering the activity	208
C.2 Algorithmic solutions developed by students	213
C.3 Algorithmic and interaction strategies by schema	230
D Pilot study with the virtual CAT	233
D.1 Screens of the final application	233
D.2 Algorithmic and interaction strategies by schema	241
Bibliography	246
Research contributions	283
Scientific publications	283
Datasets	284
Other research outputs (software, protocols, and materials)	285

Figures

3.1	The CT-cube.	33
4.1	Components of a CTP.	37
4.2	Template for defining components and characteristics of a CTP.	38
4.3	Taxonomy of CT competencies.	42
5.1	Example of BN-based learner model.	51
5.2	A noisy gate explicit formulation.	53
5.3	Example of BN modelling a task-specific assessment rubric.	58
6.1	Sequence of cross array schemas.	63
6.2	Experimental settings (unplugged CAT).	64
6.3	The algorithm activity of the CT-cube.	65
6.4	The possible cells in the algorithm activity of the CT-cube for the CAT.	66
6.5	Components and characteristics (unplugged CAT).	67
6.6	Activity profile (unplugged CAT).	69
6.7	Assessed cells in the algorithm activity of the CT-cube for the CAT.	71
6.8	Examples of algorithms for Schema 3.	72
7.1	Experimental settings (virtual CAT)	77
7.2	Components and characteristics (virtual CAT).	80
7.3	Activity profile (virtual CAT).	82
7.4	Assessed cells in the algorithm activity of the CT-cube for the virtual CAT.	82
7.5	UX design life cycle.	86
7.6	Design and evaluation process overview.	86
7.7	Example of a cross-board with coordinate labels.	91
7.8	Example of movement on the cross-board.	92
7.9	Example of colouring a row of six dots.	92
7.10	Example of repetition of a square pattern.	94
7.11	Example of cells mirroring.	94

7.12 Database schema.	97
7.13 First prototype of the CAT-VPI.	99
7.14 First prototype of the CAT-GI.	99
7.15 Second prototype of the CAT-VPI.	102
7.16 Second prototype of the CAT-GI.	102
7.17 Final CAT-VPI.	104
7.18 Final CAT-GI.	104
8.1 Example of a noisy gates BN modelling the unplugged CAT assessment rubric.	109
8.2 Example of a constrained noisy gates BN modelling the unplugged CAT assessment rubric.	110
8.3 Example of a constrained noisy gates BN modelling the unplugged CAT assessment rubric including supplementary skills.	112
8.4 The inhibition parameters for the unplugged CAT.	114
8.5 The inhibition parameters for the unplugged CAT in schema T3.	114
9.1 Age-wise distribution of CAT score levels (unplugged CAT).	121
9.2 Age-wise distribution of CAT score levels by gender (unplugged CAT).	122
9.3 Algorithmic and interaction strategies across age (unplugged CAT).	123
9.4 Algorithmic and interaction strategies across age and gender (unplugged CAT).	124
10.1 Algorithmic and interaction strategies across age (virtual CAT - pilot).	133
11.1 Age-wise distribution of interaction dimensions (virtual CAT - main).	139
11.2 Age-wise distribution of CAT score levels (virtual CAT - main).	141
11.3 Algorithmic and interaction strategies across age (virtual CAT - main).	143
11.4 Restarts distribution per interaction dimension.	144
11.5 Restarts distribution per age.	145
12.1 Gender-related school performance variations.	155
12.2 Individual student performance variations.	156
12.3 Session-Grade performance variations.	157
12.4 Schema-based performance variations.	158
12.5 Algorithm dimension variations across age categories at schema level.	159
12.6 Performance variations across age categories at schema level.	159
12.7 Task completion time and performance variations.	160
13.1 Comparison of BN-based and average CAT scores (unplugged CAT).	165

13.2 Comparison of BN-based and average CAT scores (unplugged and virtual CAT).	171
C.1 Algorithms observed for schema S1.	213
C.2 Algorithms observed for schema S2.	214
C.3 Algorithms observed for schema S3.	215
C.4 Algorithms observed for schema S4.	216
C.5 Algorithms observed for schema S5.	217
C.6 Algorithms observed for schema S6.	218
C.7 Algorithms observed for schema S7. (Continued on the next page).	219
C.7 Algorithms observed for schema S7. (Continued from the previous page).	220
C.8 Algorithms observed for schema S8. (Continued on the next page).	221
C.8 Algorithms observed for schema S8. (Continued from the previous page).	222
C.9 Algorithms observed for schema S9.	223
C.10 Algorithms observed for schema S10. (Continued on the next page).	224
C.10 Algorithms observed for schema S10. (Continued from the previous page).	225
C.10 Algorithms observed for schema S10. (Continued from the previous page).	226
C.11 Algorithms observed for schema S11. (Continued on the next page).	227
C.11 Algorithms observed for schema S11. (Continued from the previous page).	228
C.12 Algorithms observed for schema S12.	229
C.13 Algorithmic and interaction strategies across age for S1.	230
C.14 Algorithmic and interaction strategies across age for S2.	230
C.15 Algorithmic and interaction strategies across age for S3.	230
C.16 Algorithmic and interaction strategies across age for S4.	231
C.17 Algorithmic and interaction strategies across age for S5.	231
C.18 Algorithmic and interaction strategies across age for S6.	231
C.19 Algorithmic and interaction strategies across age for S7.	231
C.20 Algorithmic and interaction strategies across age for S8.	231
C.21 Algorithmic and interaction strategies across age for S9.	232
C.22 Algorithmic and interaction strategies across age for S10.	232
C.23 Algorithmic and interaction strategies across age for S11.	232
C.24 Algorithmic and interaction strategies across age for S12.	232
D.1 Language selection.	234
D.2 Module selection.	234
D.3 Training module.	235
D.4 Session form in the validation module.	236
D.5 Student form in the validation module.	236
D.6 CAT visual programming interface (CAT-VPI) with textual commands.	237

D.7 CAT visual programming interface (CAT-VPI) with symbolic commands.	237
D.8 CAT gesture interface (CAT-GI).	238
D.9 Results dashboard.	239
D.10 Pupil feedback survey.	240
D.11 Algorithmic and interaction strategies across age for S1.	241
D.12 Algorithmic and interaction strategies across age for S2.	241
D.13 Algorithmic and interaction strategies across age for S3.	242
D.14 Algorithmic and interaction strategies across age for S4.	242
D.15 Algorithmic and interaction strategies across age for S5.	242
D.16 Algorithmic and interaction strategies across age for S6.	243
D.17 Algorithmic and interaction strategies across age for S7.	243
D.18 Algorithmic and interaction strategies across age for S8.	243
D.19 Algorithmic and interaction strategies across age for S9.	244
D.20 Algorithmic and interaction strategies across age for S10.	244
D.21 Algorithmic and interaction strategies across age for S11.	244
D.22 Algorithmic and interaction strategies across age for S12.	245

Tables

1.1	Swiss compulsory education system according to the HarmoS Agreement.	3
4.1	Core skills definition (level 1).	43
4.2	Problem setting sub-skills definition (level 2 and 3).	43
4.3	Algorithm sub-skills definition (level 2 and 3).	44
4.4	Assessment sub-skills definition (level 2 and 3).	45
4.5	Profiling template for CTPs.	47
5.1	Example of a task-specific assessment rubric.	55
6.1	CAT score (unplugged CAT).	74
7.1	Differences between the unplugged and virtual CAT.	79
7.2	Differences between the unplugged and virtual CAT characteristics.	81
7.3	CAT score (virtual CAT).	84
7.4	CAT programming language commands index.	93
8.1	Definition of the CAT assessment rubric.	108
8.2	Example of answer encoding for 1D-VS in the unplugged CAT.	111
9.1	Study participants (unplugged CAT).	117
9.2	Participation and success rates across schemas (unplugged CAT).	120
9.3	Pairwise comparison of CAT scores between age groups (unplugged CAT).	122
9.4	Algorithm dimensions distribution across schemas (unplugged CAT).	125
9.5	Algorithms distribution across schemas.	126
10.1	Study participants (virtual CAT – pilot study).	128
10.2	Activity completion time across age categories (virtual CAT - pilot).	131
10.3	Time spent using each interaction dimension (virtual CAT - pilot).	131
10.4	Participation and success rates across schemas and age categories (virtual CAT - pilot).	132

11.1 Study participants (virtual CAT - main).	135
11.2 Activity completion time across age categories (virtual CAT - main). . . .	138
11.3 Time spent using each interaction dimension (virtual CAT - main).	139
11.4 Participation and success rates across schemas (virtual CAT - main). . . .	140
11.5 Participation and success rates across age categories (virtual CAT - main).	141
11.6 Pairwise comparison of CAT scores between age groups (virtual CAT - main).	142
12.1 Baseline model (M0) summary.	150
12.2 LRT to evaluate the inclusion of canton as a predictor.	150
12.3 LRT to evaluate the inclusion of gender as a predictor.	151
12.4 Model (M3) summary.	153
12.5 Type III ANOVA, with Satterthwaite's method, on model (M3).	154
12.6 Model (M4) summary.	161
12.7 Type III ANOVA, with Satterthwaite's method, on model (M4).	162
12.8 LRT to evaluate the global effect of Gender on the full dataset.	162
13.1 Comparison of inference times across models (unplugged CAT).	166
13.2 Tasks answers for a representative set of pupils (unplugged CAT).	167
13.3 Comparison of the average CAT score and the BN-based CAT scores across models for a representative subset of pupils (unplugged CAT). . .	168
13.4 Posterior probabilities for target skills across models for a representative subset of pupils (unplugged CAT).	169
13.5 Posterior probabilities for supplementary skills across models for a rep- resentative subset of pupils (unplugged CAT).	169
13.6 Comparison of inference times across models (unplugged vs virtual CAT).	171
13.7 Comparison of the average CAT score and the BN-based CAT score for the enhanced model for a representative subset of pupils (virtual CAT). .	172
13.8 Tasks answers for a representative set of pupils (virtual CAT).	173
13.9 Posterior probabilities for target skills in Model ECS for a representative subset of pupils (virtual CAT).	174
13.10 Posterior probabilities for supplementary skills in Model ECS for a rep- resentative subset of pupils (virtual CAT).	175
B.1 Comprehensive overview of the relationship between different CTP char- acteristics and CT competencies.	199

List of Acronyms

ANOVA	Analysis of Variance
AT	Algorithmic Thinking
BCTt	Beginner Computational Thinking test
BH	Benjamini-Hochberg
BKT	Bayesian Knowledge Tracing
BN	Bayesian Network
BPA_t	Basic Programming Abilities test
CAT	Cross Array Task
cCTt	competent Computational Thinking test
CPT	Conditional Probability Table
CS	Computer Science
CT	Computational Thinking
CT-cube	Computational Thinking cube
CTP	Computational Thinking Problem
CTt	Computational Thinking test
EMMs	Estimated Marginal Means
HCI	Human-Computer Interaction
HG	HarmoS grade
HMM	Hidden Markov Model
HSD	Honestly Significant Difference
IAS	Intelligent Assessment System
IRT	Item Response Theory
ITAS	Intelligent Tutoring and Assessment System
LRT	Likelihood Ratio Test
OLS	Ordinary Least Squares
PPO	Proximal Policy Optimization
R2T2	Remote Rescue with Thymio II
REML	Restricted Maximum Likelihood

(Continued)

RL	Reinforcement Learning
SCM	Structural Causal Model
STEM	Science, Technology, Engineering and Math
T&E	Trial and Error
UX	User Experience
VE	Variable Elimination

Part I

Introduction and Related works

Chapter 1

Introduction

This chapter outlines the context and objectives of this doctoral research, presenting key challenges and ethical considerations related to the study.

1.1 Research context

This doctoral research is part of a project funded by the Swiss National Science Foundation (SNSF) under the National Research Programme “Digital Transformation” (NRP 77). This programme investigates the interrelationships and specific effects of digital change in Switzerland, with a focus on understanding its impact on various sectors, including education. Specifically, the project is situated within the “Education, Learning and Digital Change” module, which examines how digitalisation influences educational content, skill acquisition, and lifelong learning processes while also identifying challenges and strategies to manage the transformation of the education system.

1.1.1 Switzerland’s education system

Switzerland’s education system, characterised by its decentralisation and multilingual environment, which includes four official languages – German, French, Italian, and Romansh –, ensures that education is tailored to the needs of each canton while maintaining a degree of coherence through the Intercantonal Agreement on Harmonisation of Compulsory Education, known as HarmoS Agreement [286, 308].

The structure of compulsory education typically spans 11 years, beginning at age 4 and ending at 15 (16 for students who repeat a year), and is divided into preschool, primary, and lower secondary levels, with subtle variations across linguistic regions (see Table 1.1). The system offers a broad curriculum at all levels that covers essential subjects such as languages, mathematics, natural sciences, and physical education.

Table 1.1 – Swiss compulsory education system according to the HarmoS Agreement. For each linguistic region – German (DE), French (FR) and Italian (IT) – are shown the stages of education, represented by three key educational cycles (preschool, primary, and lower secondary), along with the corresponding HarmoS Grades (HGs) and ages.

HG	0	1	2	3	4	5	6	7	8	9	10	11
Age	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15
DE		Kindergarten		Primarschule						Sekundarstufe I		
FR		Cycle 1 (primaire)				Cycle 2 (primaire)				Cycle 3 (secondaire 1)		
IT	Scuola dell'infanzia			Scuola elementare					Scuola media			

While the overarching framework remains consistent, regional autonomy allows for variations in how specific subjects are integrated into local curricula.

Recent shifts, particularly in response to the COVID-19 pandemic, have accelerated the focus on digital competencies, now recognised as a crucial skill for students. However, the inclusion and emphasis on these competencies vary across linguistic regions, each adopting unique approaches to their development and assessment. In the German-speaking regions, the “Lehrplan 21” incorporates digital competencies alongside core subjects such as mathematics, science, and social studies [81]. In the French-speaking regions, the “Plan d’études romand (PER)” emphasises the application of technological knowledge and Computer Science (CS) principles [55]. Similarly, in the Italian-speaking canton of Ticino, the “Piano di Studio della scuola dell’obbligo ticinese” integrates digital skills to promote critical thinking, problem-solving, and creative engagement with technology [246].

In conclusion, Switzerland’s educational framework is adapting to both regional diversity and the growing significance of digital literacy, ensuring that students are equipped with the essential skills to thrive in the digital age. This ongoing evolution reflects the country’s commitment to fostering a well-rounded education system that prepares students for the challenges and opportunities of the future while respecting regional autonomy and cultural contexts.

1.2 Research motivation

With the growing impact of technology, it has become increasingly important for individuals to develop skills to effectively use and handle new technologies and engage with problem solving processes, known as Computational Thinking (CT) skills. These competencies have been recognised as fundamental in curricula worldwide and are

considered key components of students' academic and professional success. Despite the development of numerous educational approaches in recent years, a significant gap remains in terms of replicable, scalable, and easily applicable assessment tools and protocols to evaluate computational thinking skills on a large scale. In Switzerland, efforts have been made to integrate CT into education, but the country's linguistic and cultural diversity presents challenges for implementing uniform educational strategies, requiring adaptable, context-sensitive approaches. This research aims to address these challenges by developing adaptable strategies, protocols, and instruments for assessing CT skills among compulsory school students in Switzerland. The goal is to create tools that are both easy for teachers to adopt and capable of supporting semi-automated, large-scale assessments.

1.3 Research objectives

1.3.1 Project overview

The SNSF project “Assessing the Development of Computational Thinking Skills Through an Intelligent Tutoring and Assessment System” contributes to a large-scale assessment of CT competencies among Swiss students, with the following specific goals:

1. Developing an age-based competence model for CT: The project aims to define a clear and adaptable model that categorises CT competencies according to age groups, ensuring its applicability across different educational levels and contexts.
2. Creating standardised assessment instruments: A set of standardised problems will be identified and developed to assess CT skills in students of various ages, allowing for consistent and reliable measurement of CT capabilities.
3. Developing a state-of-the-art Intelligent Tutoring and Assessment System (ITAS): The project will develop an advanced probabilistic ITAS that not only assists students in solving CT problems but also measures their performance and skill development in real-time. This system will enable semi-automatic, large-scale monitoring of CT skills in classrooms across different regions.
4. Validating the framework and measuring effectiveness: The project will test the developed methodology in real classroom settings, particularly in the cantons of St. Gallen, Vaud, and Ticino, which have distinct educational approaches to CT. By collecting data on the effectiveness of the implemented strategies, the project aims to refine and validate the framework for improving CT skill development in Swiss students.

The project is a collaborative effort involving the Dalle Molle Institute for Artificial Intelligence (IDSIA USI-SUPSI), where my research is based, the Federal Institute

of Technology in Lausanne (EPFL), the University of Teacher Education of Southern Switzerland (SUPSI-DFA), the University of Applied Sciences Northwestern Switzerland (Fachhochschule Nordwestschweiz, FHNW), and other leading academic and research institutions.

1.3.2 Doctoral research focus

Within the broader framework of CT assessment, this doctoral research specifically focuses on Algorithmic Thinking (AT), a subcomponent of CT concerned with solving problems through step-by-step procedures known as algorithms. AT was chosen as the focal construct because it underpins the development of structured reasoning, decomposition, and procedural thinking, skills essential for computational problem-solving. Research indicates that AT serves as a foundation for broader CT competencies, making its assessment particularly relevant for understanding how young learners develop computational skills. By concentrating on AT, this study aims to refine assessment methods that can capture its progression and impact in early education. Unlike the overall project, which includes both tutoring and assessment components, this thesis concentrates solely on the assessment aspect. The key objectives of the research are:

1. Developing an age-based competence model for CT: Aligning with the broader project, this objective seeks to categorise CT competencies based on age, ensuring adaptability across educational settings.
2. Developing a large-scale assessment instrument for AT: Designing an activity that effectively measures students' AT skills, irrespective of age or educational background, independent of age or educational background, using the competence model established in Objective 1.
3. Developing an Intelligent Assessment System (IAS): Implementing a probabilistic system to monitor students' AT skills in real-time during problem-solving activities, enabling large-scale, semi-automated assessment without integrating tutoring mechanisms.
4. Examining AT competencies in Swiss educational settings: Testing assessment tools and the IAS in real-world classrooms to explore how AT competencies develop across different age groups in the Swiss educational landscape, and the contextual factors influencing their progression.

1.4 Research questions

This doctoral thesis is guided by the following research questions, which span both the fields of Education and Computer Science (CS):

- RQ1. How can a competence model for CT be defined to assess skills across different age groups and educational contexts? (Education)
- RQ2. How can an activity and related instruments be developed to assess AT competencies on a large scale across different age groups and educational contexts, and what characteristics should they have to ensure their effectiveness and validity? (Computer Science & Education)
- RQ3. How can a probabilistic IAS be designed and integrated into the instrument for assessing AT skills across different age groups and educational contexts? (Computer Science)
- RQ4. What are the key AT competencies in the Swiss educational landscape, how do they develop across age groups, and what demographic or contextual factors are associated with variations in these competencies? (Education)

By addressing these research questions, this thesis contributes to both educational research and computational assessment, bridging theoretical and practical insights from both domains.

1.5 Research challenges

Following the definition of the research objectives and research questions, several challenges have emerged during this doctoral project, significantly shaping its scope and methodology.

1.5.1 Diversity of the population

First, the diversity of the Swiss educational landscape presented a complex challenge. The project aimed at assessing AT skills across all compulsory school levels in Switzerland required careful navigation of regional, linguistic, and pedagogical differences. Such elements affect not only the delivery of educational content but also the design and implementation of assessment tools, which need to be adaptable to different contexts. The heterogeneity of the student population, spanning different ages, educational settings, and activity domains, further complicates this challenge, necessitating a comprehensive approach to analysis.

To address this, we considered several factors that may influence the study's outcomes, categorising them into two main groups: demographic factors, which reflect the individual and social characteristics of the participants, and contextual factors, which relate to the environment in which the study takes place and are influenced by external circumstances rather than the individual participants.

Demographic factors Demographic factors are essential for achieving all research objectives, including developing the competence model, assessment instrument, and exploring results. In the context of this study, we considered three key demographic factors:

- *Canton*: Switzerland's linguistic diversity influences educational practices and students' experiences. By categorising participants based on their canton of residence, we can explore how variations in educational systems, specifically across the German and Italian-speaking regions, impact AT development and assessment.
- *Gender*: While gender differences do not directly influence the development of the competence model or assessment instrument, they are crucial for analysing findings. Considering gender helps identify possible disparities in AT performance and engagement, ensuring a comprehensive understanding of student experiences.
- *Age category*: The broad age span of participants (3–16 years) necessitates an assessment tool that is both versatile and developmentally appropriate. To ensure suitability across cognitive stages, we categorised students into four age groups: 3–6, 7–9, 10–13, and 14–16 years old. This categorisation was informed by both psychological and educational considerations. Developmentally, these groups align with cognitive stages outlined in Piaget's theory [223], ensuring that assessments correspond to typical cognitive milestones. Additionally, they map onto the Swiss educational cycles, covering early childhood education, primary education, and secondary education. This dual alignment allows for a structured analysis of AT skill development across key educational transitions.

Contextual factors Contextual factors primarily contribute to the objectives related to testing the assessment instrument and exploring AT competencies. These factors provide a broader and more precise context for data analysis by considering the educational settings in which students interact with AT content. The key contextual factors examined in this study include:

- *Educational context*: The study takes into account the different school types (preschool, primary, and lower secondary education) and HGs, which define the 11 specific levels within Switzerland's federal education system. Additionally, the influence of individual schools and class sessions is considered, recognising that each school has distinct student compositions, resources, and teaching strategies that may affect AT learning and assessment.
- *Activity domains*: The format in which educational activities are conducted significantly impacts student engagement and performance. We differentiate between unplugged and virtual formats, each offering distinct interaction methods

that influence the development of the competence model and assessment instrument. These formats also shape students' experiences and learning outcomes, making them a crucial factor in the study's broader analysis.

Understanding the interplay of these demographic and contextual factors allows for a more nuanced analysis of AT competence development across Switzerland's diverse educational landscape. The study's methodological approach ensures that these factors are systematically accounted for, enhancing the validity and applicability of the findings.

1.5.2 Sample size considerations

Given the diversity of the population, estimating an appropriate sample size is essential to ensure meaningful and reliable results. In this research, we referred to a simulation study by Pan et al. [216] to estimate the number of participants needed to detect a mediation effect in our analysis. Based on effect size considerations, we estimated that a sample size of 200 to 350 participants would provide an 80% probability of detecting meaningful effects.

Given the exploratory nature of this study, our primary goal was to assess the applicability of our assessment instrument across diverse age groups and educational contexts, rather than to test predefined hypotheses requiring formal power calculations. Consequently, we designed our study to include approximately 300 participants, accounting for an expected 20% attrition rate due to factors like disengagement, logistical constraints, unsigned consent forms, or unforeseen circumstances, like absenteeism on the day of the activity.

While power considerations may be relevant in the context of our statistical analysis, since we employed hierarchical linear modelling to account for the nested structure the data, the chosen modelling approach helped mitigating potential limitations related to sample size by appropriately handling variability across groups. Given the diversity of our sample and its alignment with previous studies of similar scope, we considered our sample size sufficient to detect meaningful trends and effects.

By addressing the challenges posed by population diversity, this study ensures that the assessment instrument is both robust and adaptable, capable of providing valuable insights into AT development across Switzerland's educational system.

1.5.3 Impact of the COVID-19 pandemic

Another major challenge was the unprecedented outbreak of the COVID-19 pandemic, which significantly disrupted our plans for direct engagement with schools and participants. My doctoral research began in November 2020, and we faced difficulties access-

ing school classes to conduct the necessary experimentation during the initial stages. During the first experimental study, conducted between March and April 2021, restrictions were still in place, severely impacting our ability to engage with participants in person. As a result, certain activities, such as co-designing activities in the early phase of the study, were not feasible. Additionally, during the data collection phase, we had to take extra precautions to ensure safety. These included measures such as wearing masks, maintaining physical distance, and sanitising materials regularly. While these adjustments allowed us to continue the research, they introduced constraints on the types of activities that could be conducted and required continuous attention to health protocols.

1.5.4 Ethical considerations

A final critical challenge in this study relates to the ethical responsibility of conducting research with young and vulnerable participants. To ensure participant protection, data confidentiality, and the integrity of the research process, rigorous protocols were developed in compliance with both national and international ethical standards [14, 222]. This study adhered to the EPFL Human Research Ethics Committee's (HREC) ethical guidelines and received approval (HREC No: 048-2023).

Informed consent and participant information procedures

The informed consent and participant information procedures were designed to ensure clarity and transparency for all involved parties. First, school directors and teachers were provided with an information sheet detailing the research project, the involved institutions and researcher, the experimental activities to be performed in the classrooms, as well as the nature of the data being collected, the data storage and access methods. School directors and teachers were then asked to provide explicit authorisation for the research team to access the school and classrooms for the study. Parents were also provided with the same information sheet, along with the contact details of the principal investigator, for any further inquiries. They were asked to sign and return the consent form attached to the information letter. The participant information sheet template and parental consent form are in Appendix A.

In line with ethical standards, informed consent was obtained from all participants' parents or legal guardians. Only pupils with explicit consent were allowed to participate in the study. In addition, children were informed about the study before their participation and were given the option to decide whether to participate.

Data collection and storage

The data collected in this research include session details (e.g., canton, school, class), pupil information (e.g., gender, month of birth), and performance data. In compliance with Swiss and international guidelines, data were pseudonymised to protect participant identities by eliminating identifiable information, such as assigning unique indices to schools instead of using their actual names.

All data storage and communication channels are encrypted to maintain data integrity and confidentiality. To ensure data security during transmission, secure procedures were followed to transfer data to the local server.

Access to confidential data is restricted to authorised researchers affiliated with the project, and all data are stored on the partner research institutions' Swiss servers (Switch Drive), with access controlled through secure login credentials. In line with long-term preservation, data are also stored indefinitely on Zenodo, a secure digital repository, ensuring open access and continued availability.

1.6 Research contribution

This dissertation contributes to the fields of Artificial Intelligence (AI), Human-Computer Interaction (HCI), and Education.

- This work can advance the application of AI-based assessment in education by developing and evaluating a tool for measuring AT skills. The study demonstrates how AI can support automated evaluation, offering insights into student performance across various educational stages. It contributes to AI research by refining assessment methodologies, integrating AI techniques in educational evaluation, and exploring their implications for adaptive learning systems.
- The research contributes to HCI by employing a user-centred design approach to ensure that the AI-based assessment tool is accessible, usable, and pedagogically effective. Through iterative prototyping and participatory design, the study examines how interaction design principles influence digital assessment environments. The findings provide insights into usability, engagement, and the role of intelligent systems in facilitating AT development.
- At its core, this dissertation addresses a critical gap in educational assessment by proposing a structured framework for evaluating AT across different cognitive and educational stages. By aligning the assessment tool with developmental theories and educational curricula, the study contributes to understanding how AT skills evolve over time, how digital tools can support formative assessment, and how AI-driven approaches can complement traditional evaluation methods.

By integrating AI, HCI, and educational research, this dissertation highlights the potential of AI-powered assessment tools to enhance learning processes while maintaining usability and accessibility through HCI principles. The study also critically reflects on the varying depth of exploration in each domain, emphasising their intersection and collective impact on the broader field of educational technology and digital assessment.

1.7 Structure of the thesis

The thesis is structured in five parts. Part I includes the introduction and literature review, providing an overview of the study's context, objectives, and key concepts, followed by a discussion of previous research relevant to the topic. Part II outlines the methodological frameworks developed for the research, including the age-based competence model for CT. Part III presents the instruments designed for assessing AT and the developed IAS. Part IV presents the results from the experimental studies, as well as from the application of the IAS to the collected data. Finally, Part V offers a discussion, addressing the research questions and concluding with an exposition of the study's limitations and suggestions for future work.

Chapter 2

Related works

This chapter provides a comprehensive overview of CT in education, beginning with its significance and the challenges in defining CT skills. It then delves into AT as a core component of CT and explores the integration of CT and AT in educational contexts. The chapter continues by examining existing assessment tools and the challenges involved in assessing AT, before discussing IAS and the role of probabilistic graphical models in skill assessment.

2.1 Computational Thinking in education

In an era defined by rapid technological advancement and increasing digitalisation, CT has gained significant attention across educational sectors as a foundational skill enabling students to engage with complex systems, address interdisciplinary challenges, and meet the demands of an evolving digital landscape [150, 327, 335, 338]. Over the past two decades, substantial research on various facets of CT has underscored its importance as a 21st-century competence, driving global efforts to integrate it into K-12 education [327].

The term CT was popularised by Jeannette Wing, who described it as “*a fundamental skill for everyone, not just for computer scientists*” [335]. She proposed that CT should be integrated into every child’s analytical toolkit alongside reading, writing, and arithmetic. In particular, she defined CT as “*the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out*” [338]. Wing underscored that CT extends beyond computational tools, emphasising key processes such as abstraction, decomposition, and algorithmic design.

As highlighted by Rapaport [237], the term *CT* comprises two essential components. The first, “computational”, implies a focus on computation, which in turn in-

volves the design and execution of algorithms. The second, “thinking”, reflects the cognitive process required for problem-solving. This perspective aligns with Wing’s view, suggesting that CT is inherently tied to algorithms rather than artificial agents (e.g., computers) [338]. Consequently, CT encompasses both digital and unplugged tasks, where the solution involves an algorithm to accomplish a specific task. CT is thus defined as a cognitive process involving interrelated skills such as *decomposing* complex problems into manageable components, identifying *patterns* and recurring structures, *abstracting* relevant information from irrelevant details, and formulating *algorithms* to devise effective solutions [77, 78, 175, 217, 218, 255, 294, 335, 336, 337].

2.2 Algorithmic Thinking in education

AT is a specific aspect of CT, focusing on the design of step-by-step procedures or algorithms to solve problems systematically and achieve specific outcomes [99, 175, 255, 287]. While CT broadly encompasses skills like decomposition, pattern recognition, and abstraction, AT is more narrowly concerned with the development of algorithms, which are crucial for structured problem-solving in both human and computer contexts [25, 99, 233, 270].

In educational settings, AT is recognised as a critical skill for fostering logical reasoning, creativity, and problem-solving [18, 147, 342, 346]. These skills extend beyond the realm of computer science (CS) to address broader interdisciplinary challenges, making AT an essential competency in the digital world across personal and professional domains [33, 38, 169, 325, 327, 334, 335]. Its integration into education has gained significant momentum, as it equips students with the ability to break down complex problems, devise sequential actions, structure systematic solutions, and comprehend foundational concepts like algorithms and data structures [72, 162, 165, 213, 214].

The theoretical underpinnings of AT draw on early developmental theories from Piaget and Vygotsky, emphasising the role of active learning and social interactions in constructing knowledge during the early stages of childhood development [223, 224, 225, 319]. Piaget’s constructivist theory posits that children build knowledge through hands-on experiences and interactions with their environment, while Vygotsky’s social constructivist theory adds that social interactions and cultural context significantly influence learning outcomes. These perspectives align with the idea that engaging students in problem-solving and critical-thinking activities, such as those involved in AT, can significantly enhance cognitive development.

Modern research expands on these foundations, underscoring the importance of introducing AT concepts early in education to cultivate critical thinking, logical rea-

soning, and analytical skills [101, 138, 146, 154, 208, 317]. These skills are not only essential for STEM education but also have transferable applications across various disciplines [33, 38, 325, 327]. This growing emphasis reflects a broader understanding of education that values skills relevant to the demands of the 21st century [93, 228, 264, 315].

2.3 Defining Computational Thinking

Defining CT and AT is a significant challenge in both research and practice despite the growth in tools, activities, and curricula designed to teach it. The absence of a universally accepted definition of CT and its relationship to AT represents an obstacle to its integration into educational standards and curricula [327]. Various definitions have emerged, each emphasising different aspects of CT, which has hindered the development of a coherent framework for developing and assessing CT and AT competencies [171]. As a result, the field has struggled to move beyond an exploratory stage.

Unlike other introductory skills such as reading, writing, and arithmetic, research on CT and AT has not consistently considered its developmental nature. Specifically, existing literature rarely considers age or developmental progression when defining or discussing these competencies [275, 299]. Additionally, complex components such as abstraction, a key element often associated with CT [275, 338], are considered beyond the cognitive capabilities of very young individuals. Moreover, many existing CT models primarily focus on internal cognitive processes, neglecting the situated nature of tasks that require CT, such as the social context and the artefactual environment in which these tasks occur.

The field has also struggled to establish a structured approach for assessing CT and AT due to the complexity of its components. Efforts to decompose CT into sub-dimensions, such as decomposition, generalisation, and pattern recognition [43, 115, 255], have encountered difficulties because these dimensions are often interwoven, making it challenging to assess them independently [171, 255].

As a result, there is a pressing need for a more structured and precise approach to defining CT and AT. Establishing a clear, comprehensive, and standardised definition would not only clarify the concept but also facilitate the design of effective interventions and assessment tools. Additionally, this definition should explicitly consider the developmental nature of cognitive abilities, the age of learners, and the context in which CT and AT activities are situated. Such an approach would provide a more accurate foundation for understanding how these skills evolve throughout an individual's learning journey. This, in turn, would support the long-term integration into educational curricula, enabling its development and assessment in a consistent and meaningful

way.

2.3.1 A situated cognition perspective

The most widely recognised definitions of CT often emphasise the cognitive processes an individual should activate to solve a computational task [275]. These definitions typically align with the classical view of cognition as an internal process occurring within a single individual, i.e., the “thought processes” in the definition of Wing [338].

An alternative perspective on CT emerges from theories of situated cognition, which view cognitive activities as inherently social and contextual [125, 258]. According to this view, cognitive activities are not isolated within the individual’s mind but are shaped by interactions with the environment and social context. This perspective suggests that learning and knowledge construction occur through shared practices, where external cognitive artefacts play a crucial role in mediating thinking and problem-solving.

In educational settings, as well as more broadly, CT and AT are often activated in environments that involve social interactions and rich artefactual contexts, aligning better with this situated approach to cognition. Theories of situated learning further support this notion, highlighting that learning is most effective when it occurs in authentic, meaningful contexts, where knowledge is co-constructed through engagement with the environment and community rather than through abstract, isolated instruction [125, 258]. In the case of AT, this perspective stresses that algorithmic problem-solving is often influenced by the surrounding context, tools, and social interactions that learners engage with.

2.3.2 A developmental perspective

Mathematical thinking is a multifaceted set of skills and attitudes widely recognised as a fundamental component of human thinking that evolves over time, beginning early in life and developing throughout education into adulthood [291]. A foundational example of this progression is the act of counting, which starts with basic enumeration and evolves into more complex concepts like the idea of “number” [74, 100]. Pupils continuously refine these skills through various experiences ranging from their first counting experiences with concrete objects in pre-primary school and even before [32, 45, 261], through the development of counting strategies, symbolisation, automatisisation, abstraction, and so on, up to axiomatisation of natural numbers and the development of very complex counting algorithms.

Tall’s model of the three worlds of mathematics provides a framework to describe how mathematical thinking develops over time [290, 291, 292]. The model suggests

that acquiring mathematical concepts and theories unfolds incrementally, starting with concrete, embodied experiences, moving through internalisation via symbolic concepts (procepts), and culminating in abstract axiomatisation. Tall argues that mathematical thinking progresses through three interconnected stages:

- *an object-based conceptual-embodied world reflecting on the senses to observe, describe, define and deduce properties developing from thought experiment to Euclidean proof;*
- *an action-based proceptual-symbolic world that compresses action schemas into thinkable concepts operating dually as process and concept (procept);*
- *a property-based formal-axiomatic world focused on building axiomatic systems based on formal definitions and set-theoretic proof.”*

Each stage builds upon the experiences from the previous one, indicating a developmental progression rather than isolated stages. For example, in the development of counting and the concept of number, these stages can be mapped to various educational levels: in pre-primary school, reasoning is primarily rooted in the first world; in primary school, both the first and second worlds are present; in secondary school, reasoning shifts predominantly to the second world, with the introduction of the third world; and at the tertiary level, reasoning is primarily situated in the third world.

This developmental model of mathematical thinking closely parallels the progression of AT. Just as mathematical concepts evolve from concrete experiences to abstract formalisation, AT follows a similar trajectory:

- Concrete execution: Young children engage in sequential actions, such as counting objects or following step-by-step instructions;
- Symbolic representation: As they develop, they learn to represent these sequences using symbols, diagrams, or structured notation;
- Abstract generalisation: Eventually, they internalise algorithmic structures, enabling them to design, analyse, and optimise problem-solving procedures independently of specific contexts.

In Tall’s framework, the transition from the conceptual-embodied world to the proceptual-symbolic world mirrors the way learners move from performing concrete steps in an algorithm to recognising and manipulating these steps as symbolic entities. Similarly, the shift to the formal-axiomatic world aligns with the ability to construct, prove, and reason about algorithms in a rigorous and abstract manner.

This perspective has important implications for assessing AT: it suggests that an individual’s competence in AT should be evaluated in relation to their cognitive development, considering their ability to engage with different levels of abstraction. Certain algorithmic constructs may be too complex for younger or less experienced learners, just as advanced mathematical reasoning requires foundational cognitive skills developed over time. Thus, the developmental trajectory of mathematical thinking provides

a valuable lens for understanding how AT emerges and evolves, informing both educational strategies and assessment frameworks for computational competencies.

2.4 Computational Thinking and Algorithmic Thinking integration in education

2.4.1 Global overview

In the *United States*, the emphasis on CT began in the early 2000s, particularly through initiatives like the Next Generation Science Standards (NGSS) [46, 205, 266, 330] and the CS for ALL Students initiative [66, 136], which aimed to ensure CS education is accessible to all students from early education onward. Other countries are also advancing in this area. *New Zealand, Australia, South Korea* and *Japan* from 2015 have started integrating digital technologies, CT and AT across STEM subjects at all educational levels, focusing on themes like algorithms and problem-solving, making programming a compulsory subject [1, 37, 194, 219, 296]. Similarly, *Singapore*, under the 2014 Smart Nation initiative led by the Prime Minister to promote early programming exposure [130, 172], launched a CT framework in 2016, introduced a computing subject focused on programming and algorithms in secondary schools by 2017, and mandatory CT and coding program for upper primary students by 2020 [37, 38, 58]. In *Canada* (British Columbia), CT has been incorporated into middle school subjects, with plans for broader application at the secondary level [37, 44].

2.4.2 European context

Several European countries have significantly advanced in integrating CT and AT into their compulsory education systems. While some have incorporated these skills across all compulsory educational levels, others have focused primarily on secondary education. The degree of integration and the scope of the curricula reforms vary widely across the continent, with some countries adopting a holistic, cross-curricular approach, while others emphasise CS or technology education as separate subjects [37, 38].

The pioneers in integrating CT and AT across both primary and secondary levels have significantly influenced the approaches of subsequent nations. Among them, *England* was one of the earliest to make CT mandatory, incorporating it into its national curriculum in 2014 as a separate subject [305]. *France* followed closely, integrating CT within existing subjects such as mathematics and technology in 2015 [98]. *Finland* incorporated CT and AT in 2016 as a cross-curricular theme, later extending its integration within subjects like mathematics, crafts, and environmental studies by 2022

[95].

In the years following these initial pioneering efforts, several other countries have embraced CT and AT, albeit at different rates and in various formats. Countries such as *Malta, Slovakia, Poland, Portugal, Croatia, Greece, Austria, and Hungary* have integrated CT/AT as a separate subject, primarily through informatics or CS courses, emphasising the importance of computational skills as a distinct area of study with dedicated instructional time [134, 140, 151, 167, 192, 209, 259, 288, 307]. In contrast, *Sweden, Norway and Lithuania* have opted to embed CT within existing subjects, such as mathematics, science, and the arts, promoting an interdisciplinary model that fosters CT across various academic domains [71, 92, 126]. In *Cyprus, Luxembourg, and Serbia*, CT is integrated into primary education primarily within other subjects, while in secondary education, it is structured as a separate subject, reflecting a flexible and context-specific approach to embedding CT across different educational levels [69, 84, 187].

Despite notable advancements in various countries, several have achieved only partial integration of CT and AT. Specifically, *Ireland, Romania, and Scotland* have incorporated these skills into secondary education, while formal integration at the primary level continues to be lacking [85, 107, 128, 297].

Additionally, several countries have made little to no progress in integrating CT and AT into their educational systems. In *Denmark, Slovenia, Italy, the Czech Republic, the Netherlands, and Spain*, the situation varies, with most of these countries at the drafting stage of curricula or strategic plans for future actions [37, 38, 49, 70, 111, 193, 247]. For instance, Denmark has yet to integrate CT but has initiated a pilot program [328], while Italy recognises CT as a key topic but lacks formal integration in its national curriculum.

The situation in Belgium further illustrates this complexity, as integration depends on specific regions. In Flanders, CT has been integrated as part of a separate subject, while Wallonia plans to address it as a compulsory subject for primary and lower secondary schools [64].

2.4.3 Swiss context

The Swiss educational system has progressively integrated AT and CT into its curriculum, adapting to the specific needs of its diverse linguistic regions. These skills are embedded within various subjects, such as mathematics and CS, through activities like coding, algorithmic exercises, and robotics projects, ensuring that students acquire essential skills from an early age [37, 38].

In the German-speaking region, the integration of CT began around 2014, with competencies such as coding and programming incorporated into the curricula of pri-

mary and lower secondary schools. At the upper secondary level, these skills are formalised within the national curriculum framework for non-vocational schools, ensuring a comprehensive acquisition of computational skills throughout the educational journey [79, 81]. In the French-speaking region, CT is taught through the Plan d'études romand (PER) under the subject MITIC (Média, Image, Technologie de l'Information et de la Communication), implemented since 2015 [55]. Within the framework, students engage in activities that require them to analyse problems, devise solutions, and implement basic programs, reinforcing CT skills from early education onward. Additionally, the subject "Media and Informatics" introduces CT as a core component, fostering logical thinking and problem-solving abilities. In the Italian-speaking region, AT and CT are integrated primarily through subjects like mathematics, with a strong emphasis on coding, problem-solving, and robotics. However, computer science is considered a transversal competency rather than a distinct subject. This approach encourages students to apply these skills across various disciplines, promoting interdisciplinary learning. While this may give the impression of a highly technological approach, the reality is that CT and AT are woven into the curriculum in a way that emphasises their applicability in different contexts rather than as standalone subjects [245, 246].

2.5 Assessing Algorithmic Thinking

With growing recognition of AT as an essential component of compulsory education, there is an increasing need for reliable, scalable assessment instruments that can measure students' development across various age groups and educational settings on a large scale [25, 114, 171, 210, 316].

Research on AT assessment is limited and often contradictory [113, 293, 299]. Many assessment tools exist, but each focuses on specific aspects, and none cover all educational and cognitive needs. Current methods often neglect developmental stages, social contexts, and available resources [43, 165, 166, 171, 230, 252, 253, 255, 256, 300, 342]. The lack of standardised tools and diverse evaluation methods make it difficult to assess students' overall progress [93, 97, 112, 211, 228, 264].

Empirical research has examined the effectiveness and challenges of various instruments used to assess AT. Traditional methods, such as written tests and multiple-choice questions, are widely used for their efficiency in covering broad topics, straightforward administration and grading; however, they may oversimplify the assessment by focusing on rote memorisation rather than deeper problem-solving skills [47, 65, 215, 276, 329]. In contrast, open-ended tasks and problem-solving exercises offer richer evaluations by assessing reasoning and creativity, but they can be more challenging to grade [65].

Unplugged methods, which involve hands-on, tangible activities, assess AT concepts effectively, particularly in environments without access to basic technology infrastructure [27, 42, 75, 152, 242, 339]. These activities are especially useful for building a strong foundation in computational principles and for young children without prior programming experience [42, 309, 321]. Empirical evidence from Relkin et al. [242] and Brackmann et al. [42] supports the unplugged approach, showing improvements in students' CT skills after participating in unplugged computing instruction. Moreover, studies by Brackmann et al. [41], Delal and Oner [76], Tsarava et al. [301, 302] have highlighted the positive effects of unplugged activities on motivation, engagement, and overall effectiveness, particularly in primary education. Research by Del Olmo-Muñoz et al. [75], Saxena et al. [262] has shown that unplugged activities can significantly enhance CT skills. Further, Delal and Oner [76] cited various studies confirming the development of CT skills through unplugged computing activities while also improving students' understanding of CS concepts and fostering greater interest in the subject [28, 61, 127, 176, 190, 248]. Specifically, Relkin et al. [242] found that unplugged programming activities help students achieve the highest levels of understanding in AT, logic predictions, and debugging concepts, further solidifying the effectiveness of unplugged methods in fostering foundational computational thinking. However, unplugged activities are not ideal for large-scale assessments due to their hands-on, time-intensive nature, which can be challenging to scale for large groups of students and may require significant resources and coordination [75, 90].

Digital methods, including programming assignments, coding challenges, and other computer-based activities, engage students in practical applications of algorithmic skills through interactive and individualised experiences. While these methods are scalable and well-suited for large-scale assessments, they often require intensive grading efforts and may lack the physical engagement provided by unplugged activities [243, 256, 281, 348]. Additionally, reliance on technology can pose barriers for students with limited access or those who prefer non-digital learning environments [28, 42, 152, 157, 189, 243].

The current landscape of tools for assessing AT reveals a significant gap in the availability of instruments that can meet the diverse needs of learners and educational contexts. One of the key shortcomings is the lack of tools that effectively integrate multimodal features. While unimodal tools, those offering a limited artefactual environment with a single method of interaction, may serve some students, they often fail to accommodate the diverse learning styles found in modern classrooms, limiting engagement and the accuracy of assessments. In contrast, multimodal tools, which provide various options for interaction, offer greater flexibility, allowing learners to choose the method that best suits their learning preferences, thus improving both engagement and assessment reliability.

Additionally, the target age range of a tool plays a pivotal role in its effectiveness. Tools designed for narrow age groups often fail to account for the diversity of developmental stages within a typical classroom. For example, the BPA_t [199] and the cCT_t [90] are designed for specific ages, and while they provide valuable assessments of foundational skills, their applicability in projects with a broader student demographic is limited, as they do not cater to students outside of the targeted age groups. This restriction hampers their versatility in diverse classroom settings. In contrast, instruments that cater to a broader age range ensure versatility and inclusivity, making them more adaptable to different classroom settings. Tools like Scratch [116, 180, 196] and Code.org [104] are widely used in educational settings and cater to a broad age range, allowing for engagement with students across different developmental stages. Code.org, for example, provides a wide variety of coding activities that can engage students from early primary school to high school. However, while these platforms are suitable for various age groups, the specific activities they offer are often tailored to specific age ranges, which limits their flexibility for cross-age assessments. Similarly, Scratch is adaptable to many age groups but is typically used with younger learners for simpler projects, which limits its application for more advanced learners. To address this limitation, instruments that cover a wide age range and are adaptable to different developmental stages are needed. These tools are better suited for longitudinal assessments, tracking students' progress over time and offering a comprehensive understanding of how their skills evolve across cognitive stages.

Finally, another significant gap in the current tools for assessing AT is their inability to address the full range of cognitive levels in skill development. According to the frameworks of Bloom et al. [36], Gouws et al. [106], cognitive skills progress through several stages, from foundational levels to more advanced stages. The progression typically starts with basic levels, where students focus on recognising and understanding key concepts. At this stage, students may identify patterns, follow simple instructions, or understand basic algorithms. For example, tools like the Basic programming abilities test (BPA_t) [199], the Computational Thinking test (CT_t) [251], the Beginners Computational Thinking test (BCT_t) [347, 348] and the competent Computational Thinking test (cCT_t) [90] focus on foundational skills such as recognition and understanding, limiting their effectiveness for more advanced assessments. As learners advance, they move to the next level, where they apply these basic concepts in more complex contexts. This stage involves using knowledge to solve problems, create algorithms, or engage in basic programming. Tools like Scratch [116, 180, 196] and Code.org [104] support students in applying their knowledge to create projects, such as games or animations, and apply algorithms in a more interactive environment. The final stages of the progression are characterised by higher-order skills such as analysis, synthesis, and abstraction. At this stage, students have assimilated these competencies and should

be able to evaluate complex algorithms, analyse problems deeply, and create more sophisticated solutions. Unfortunately, many existing tools fail to adequately address this advanced level of thinking, limiting their ability to assess long-term progression.

2.6 Intelligent Assessment Systems

The challenges in assessing AT have been a long-standing issue in education. Traditional manual assessment methods are time-consuming, inconsistent, and difficult to scale. The time spent on assessments often outweighs the educational value, and inconsistent results lead to disparities in evaluating student performance [234, 254]. Additionally, the subjectivity of these assessments makes them unreliable and non-standardised, raising concerns about their accuracy and fairness, especially when comparing large groups of students. Furthermore, general educators often lack the technical expertise to assess the complex aspects of AT [185, 306, 342]. While some schools try to involve IT professionals in the assessment process, these experts are often unavailable in under-resourced areas. This shortage further complicates the accurate assessment of AT. As a result, educators must rely on standardised tools, which, though consistent, often oversimplify the evaluation and fail to capture the full scope of AT skills [78].

Given these challenges, IASs represent a promising solution that can improve and simplify AT evaluation in educational contexts [117, 234, 254, 279]. These systems would provide scalable, consistent, and objective assessments, addressing the limitations of traditional methods and standardised tools [234, 254]. These technology-driven assessment tools, which are developed by experts and integrated into educational platforms, can be easily administered by teachers without specialised training. They also allow students to work independently, providing real-time feedback and evaluations without the need for constant supervision by a teacher. This autonomy not only supports individualised learning but also ensures that students receive immediate insights into their progress. Furthermore, IAS can adapt to various cognitive levels and educational contexts, making them suitable for diverse learning environments and student needs. This flexibility ensures that students at different stages of learning can benefit from the system's adaptive features.

Despite the benefits of IAS, these systems are still evolving in their ability to assess complex AT skills comprehensively. While IASs excel in providing scalable, consistent feedback, they face limitations in monitoring long-term progress, particularly when evaluating higher-order cognitive skills and more nuanced problem-solving strategies [279]. These systems are particularly effective for assessing basic and intermediate levels of AT, but they struggle to capture the development of more advanced skills over

time. Ongoing research is focusing on enhancing these systems to evaluate the full spectrum of AT competencies, with a particular emphasis on tracking how students' abilities evolve. By improving their ability to measure progress over extended periods, IAS can offer more meaningful insights into student development, thus providing a more complete picture of learning outcomes in AT.

2.6.1 Learner modelling and competence profiling

IASs collect data on a learner's performance while accomplishing a task and use this data to develop a competence profile based on a predefined model of the learner's knowledge and behaviour. As new knowledge is gathered through the learning activity, the competence profile is continuously updated. This allows for more personalised and adaptive assessments, as it reflects the learner's current state and progress.

A typical modelling approach is to define a *learner model* that mathematically describes the learner's competencies, represented by a hidden variable, referred to as *skills*. These skills influence observable actions or manifest variables, which are the learner's behaviours while solving tasks. An *assessment rubric* provides a structured way to evaluate a student's performance. It typically includes a list of competence components to be assessed, qualitative descriptions of observable behaviours corresponding to various competence levels, and criteria for evaluating each component. This rubric explicitly defines the relationship between competencies and the learner's observable actions, helping to formalise and codify the evaluation process. By specifying competence levels and corresponding behaviours, the rubric effectively guides assessments and enables accurate measurement of student skills.

2.6.2 Probabilistic graphical models

Several sources of uncertainty and variability can affect the relationship between the non-observable competencies and the corresponding observable actions. As a result, a deterministic approach cannot accurately model this relationship. Instead, probabilistic reasoning provides a more appropriate method for translating qualitative assessment rubrics into standardised, quantitative measures of student proficiency [188]. This approach accounts for the inherent uncertainties and variability in student performance, enabling more precise evaluations of learner skills and competencies.

In learner knowledge modelling, common probabilistic approaches include Bayesian Knowledge Tracing (BKT) [59], Item Response Theory (IRT) [91], and Bayesian Networks (BNs) [161]. These models help estimate a learner's knowledge state based on observable data, offering a way to track learner progress and predict future performance. IRT has limitations when dealing with multiple skills, particularly

in modelling skill correlations. In such cases, more expressive probabilistic formalisms are required.

Bayesian-based learner modelling

Among these alternatives, BNs represent a highly interpretable option and are frequently adopted in the implementation of IASs [191]. In their comprehensive review, Desmarais and Baker [80] recognised and presented BNs as the most general approach to modelling learner skills, highlighting their versatility and effectiveness in educational contexts. Mousavinasab et al. [197] systematically reviewed 53 papers about IASs applications from 2007 to 2017, exploring the characteristics, applications, and evaluation methods, and found that a significant proportion of the reviewed papers employed BN techniques, highlighting their widespread adoption and success in modelling learner knowledge. More recent works continue to support the use of BNs. For instance, Hooshyar et al. [132] developed a system designed to help students acquire problem-solving skills in computer programming using a BN model to track and assess their progress. Additionally, Xing et al. [341] introduced an automatic assessment method for evaluating students' engineering design performance, leveraging BNs for real-time evaluation. Other studies, such as those by Wu [340] and Rodriguez-Barrios et al. [249], further advocate for the use of BNs in constructing IASs, demonstrating the ongoing interest in these techniques to enhance educational technologies and assessments.

BNs are directed graphical models that effectively represent complex relationships between multiple skills, enabling dynamic updates to the learner's knowledge state as new information becomes available. One of the key advantages of BNs is their ability to model causal relationships, helping to understand how different learner behaviours influence learning outcomes. The graphical nature of these models enhances their interpretability, making them accessible to domain experts who can use them to refine and elicit student models [191]. This interpretability is especially valuable in educational settings, as it allows educators and researchers to improve the design and functionality of IASs by understanding the underlying mechanics of the model.

Despite their advantages, BNs present several challenges in their design and implementation. Not all BNs are easy to design, and creating an effective model requires a deep understanding of BN theory. While BNs are often considered causal models, defining them is not always straightforward due to the complexity of the causal relationships involved and the presence of hidden causes, which complicate the causal dynamics. Additionally, obtaining the network structure and parameters can be challenging. These parameters often need to be elicited through expert knowledge or inferred from large datasets, which may not always be available. Even when the structure of the learning

model is well-defined, the task of eliciting and learning BN parameters can make the computation of inferences unmanageable. The complexity of these models increases as the number of parameters grows, particularly with the addition of more arcs to the network. A large number of parameters not only complicates the inference process but can also slow down real-time computation, which is crucial for providing timely feedback in IASs. This complexity might discourage practitioners from adopting them, as the effort involved in eliciting expert knowledge or gathering extensive datasets can be overwhelming. Additionally, managing the computational demands for real-time feedback can be a significant challenge, especially when dealing with many parameters. For these reasons, managing the number of parameters is critical in ensuring efficient processing.

To address the challenges posed by the large number of parameters in BNs, some research has focused on reducing the model's complexity. One such approach is the use of *noisy-OR gates* for a more compact parametrisation of the Conditional Probability Tables (CPTs) within the BN [21]. This method reduces the exponential complexity of parameter elicitation, transforming it from a task that scales exponentially with the number of parent skills for each observable action into a more manageable linear process [220].

To enable real-time assessment while still capturing the necessary complexity, it is essential to create learner models that are both accurate and computationally efficient. To address this challenge, general approaches have been developed for translating assessment rubrics into interpretable BN-based learner models [181]. These models aim to be sufficiently simple to allow for fast computation and real-time feedback while maintaining enough complexity to accurately reflect the relationships between the learner's skills and their observable actions. This balance ensures that the model is computationally efficient and provides meaningful insights into the learning process. Moreover, learner models based on assessment rubrics are more accessible to teachers, who are typically more familiar with them than probabilistic graphical models. Teachers can assess student competencies through realistic scenarios specifically designed for this purpose, allowing them to compare actual student performance with a model of competence outlined by an assessment rubric [73].

BN models can be extended to incorporate hierarchies of competencies to account for the acquisition order of competencies and capture complex learner behaviours. While rubrics focus on specific behaviours in context, they indirectly model the interactions between skills by organising them into a hierarchy. This allows the model to assign higher probabilities to advanced competencies without assuming that mastering a higher-level skill automatically implies mastery of all lower-level skills. By integrating the constraints defined by the rubric through auxiliary nodes, the model can maintain its simplicity while better reflecting the complexity of learner progression. This ap-

proach eliminates the assumption of direct skill acquisition and improves the accuracy of assessments [181].

Moreover, considering that assessment rubrics typically focus only on the competence components being assessed, referred to as *target skills*, this modelling approach may result in oversimplified learner models that fail to capture the full range of factors contributing to a learner's performance. Failures are not always due to the absence of target skills but may stem from deficiencies in other skills required for the specific task, known as *supplementary skills*. To address this, BN models can be extended to incorporate supplementary skills alongside target skills. This requires expanding approaches like the noisy-OR model to represent both disjunctive (OR) and conjunctive (AND) relationships between skills [21]. While the behaviours in the assessment rubric are often mutually exclusive (OR), the combination of supplementary and target skills should be modelled together (AND) to reflect the learner's ability to complete the task more accurately.

2.7 Gaps in existing research and the contribution of this study

This section addresses the gaps identified in the existing research on CT and AT in education, particularly in the context of assessing these competencies across different age groups and educational contexts. We highlight how this study aims to address these challenges and provide a more comprehensive framework for assessing and understanding the development of AT competencies.

2.7.1 Defining a competence model for Computational Thinking - RQ1

As identified in the literature, one of the key challenges in assessing CT is the lack of a standardised and widely accepted definition. This lack of a common understanding makes it difficult to establish a universal competence model that can be applied across different contexts. Existing competence models often focus on isolated skills, ignoring how different CT skills interconnect and evolve over time. Furthermore, these models tend to be designed for specific age groups, making it challenging to understand how these skills develop from early childhood through to later stages of education. Additionally, many models overlook the contextual influences on the development of CT skills, particularly the role of social interactions and the use of tools.

To address these gaps, our approach begins with the formalisation of a comprehensive definition of CT that takes into account both its cognitive and contextual dimen-

sions. This framework will provide a more holistic understanding of CT, considering not just the individual cognitive processes but also the social and contextual influences that shape its development. The details of this comprehensive model are presented in Chapter 3, which explores the CT-cube framework in depth, alongside Chapter 4, where we further elaborate on the application of this model in practice.

2.7.2 Developing assessment instruments for Algorithmic Thinking - RQ2

To effectively assess AT competencies, it is crucial to have relevant problems, activities, and instruments that are capable of capturing the complexities of AT across different age groups and contexts. However, as highlighted in the literature review, existing assessment tools are often constrained by the same limitations as traditional competence models. They tend to focus on narrow skills, are age-specific, and overlook the situated nature of AT, particularly the role of social interactions and artefacts in learning.

In response to these challenges, we decided to develop a new assessment instrument aligned with our competence model for CT. The first step in this process involved the development of an unplugged activity called the Cross Array Task (CAT), which allowed us to gain insights into the complexities involved in evaluating AT across different age groups and contexts. However, several limitations emerged from this approach, including the variability in how instructions were interpreted and coded by different human administrators, the lack of immediate feedback for students, and the time-consuming nature of the assessment.

To address these issues, we transitioned to a digital version of the CAT, known as the virtual CAT. This new version automates the interpretation and codification of instructions, providing a more standardised approach to assessment. Furthermore, the digital platform enables real-time feedback, enhancing the reflective process for students and allowing for large-scale implementation.

While the virtual CAT improves upon the unplugged version in several key areas, it still presents some limitations, such as producing a single, task-specific score that does not capture the full scope of a student's abilities. This highlights the ongoing need for a more comprehensive evaluation approach, which we aim to address through the integration of a probabilistic IAS in the next phase of our research.

Details on the development and evaluation of these assessment instruments are presented in Part III.

2.7.3 Designing a probabilistic Intelligent Assessment System for Algorithmic Thinking assessment - RQ3

The limitations of the first version of the virtual CAT assessment instruments, such as the inability to capture a full range of skills across different tasks and contexts, necessitate the use of a more dynamic, data-driven approach. To address these shortcomings, we propose the integration of a probabilistic IAS, which evaluates student performance and tracks interactions across tasks over time. This method uses probabilistic models, such as BNs, to represent a learner's competencies and assess their progress. BNs offer several advantages, including the ability to model complex relationships between skills and competencies. However, as the number of competencies increases, the computational complexity of these models grows exponentially. To address this challenge, we adopt Noisy-Gate BNs, which reduce the number of parameters required, making the model more efficient and manageable. The integration of the probabilistic IAS provides a more detailed and holistic view of a student's competence profile, offering real-time assessment and enabling long-term tracking of progress. This approach successfully addresses the completeness issue identified in the earlier assessments and provides a more comprehensive picture of student competencies. The implementation of this system is detailed in Chapter 5, which covers the definition of the IAS model, and Chapter 8, which discusses its adaptation to the CAT case.

2.7.4 Understanding Algorithmic Thinking competencies in the Swiss educational context - RQ4

Despite the growing interest in AT in education, there is a lack of comprehensive studies on the specific competencies required for AT across different educational contexts, particularly in Switzerland. This gap is significant because understanding the key AT competencies in the Swiss educational landscape will provide valuable insights into how these competencies develop across age groups and what factors, such as demographic or contextual variables, are associated with variations in these competencies.

This study aims to address this gap by conducting a thorough analysis of AT competencies in Switzerland, which could potentially be generalised to other contexts. The findings from this analysis will not only contribute to the understanding of how AT competencies evolve across age groups but also help refine the assessment instruments we have developed. This is crucial for tailoring our tools to different educational contexts and ensuring their effectiveness across diverse learning environments. The results of this analysis are presented in Part IV.

Part II

Methodological frameworks

Chapter 3

A framework for the design and the assessment of Computational Thinking activities

The content of this chapter has been adapted from the following article with permission of all co-authors and publishers:

- Piatti, A., **Adorni, G.**, El-Hamamsy, L., Negrini, L., Assaf, D., Gambardella, L., and Mondada, F. (2022). The CT-cube: A framework for the design and the assessment of computational thinking activities. *Computers in Human Behavior Reports* [227].

As an author of this publication, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation.

3.1 Summary

This chapter addresses RQ1 by proposing a competence model for CT that considers both developmental and situated aspects of CT. Specifically, it explores how models from mathematical thinking can be used to model the evolution of CT, contributing to a theoretical foundation for assessing CT skills across different age groups and educational contexts. In particular, we extend a model introduced by David Tall for the development of mathematical thinking [290, 291, 292] to conceptualise the evolution of CT skills. Furthermore, we adopt the situated cognition framework by Roth and Jornet [258], emphasising the context-dependent and developmental nature of CT. This approach allows us to account for both cognitive progression and the role of real-world learning environments in shaping CT.

As a result, we present the Computational Thinking cube (CT-cube), a novel framework that supports the design, analysis and assessment of CT activities. It extends existing CT models to provide a life-long developmental perspective, from childhood to adulthood, while incorporating the situated nature of CT, recognising that skills emerge and evolve in different educational and practical settings.

By introducing this framework, this chapter lays the theoretical foundation for defining a CT competence model (RQ1), which later informs the design of assessment tools and activities in subsequent chapters.

3.2 Definition of Computational Thinking

In our view, consistently with the original definition of Wing [338], CT is not strictly related to the presence of artificial agents (e.g., computers) but to that of an *algorithm*. We consider a CT problem to be an activity whose solution consists of an algorithm that an artificial or human agent should perform to realise a specific task, encompassing thus also unplugged tasks which require the use of algorithms to be realised.

Our formulation of CT is also inspired by Roth's basic concepts of situated cognition [258]: cognitive activities are embodied, enacted and embedded in a situated cognitive system [125], consisting of a social context and/or an artefactual environment. In other words, cognitive activities do not occur in the head of a single individual but are shared social practices, mainly based on representations and manipulation of knowledge and information through external cognitive artefacts. We argue that this theoretical setting corresponds better to the concrete settings in which CT is activated (in education but also in general), which are usually characterised by the simultaneous presence of several persons (social setting) and a rich artefactual environment. As such, we define CT as follows:

Computational Thinking (CT) is a **situated cognitive activity**, individual or collective, consisting of three, eventually iterative, steps:

- 1.1 **Problem setting:** recognising, understanding, contextualising, reformulating and/or modelling a problem within a specific context in such a way that its solution can be computed;
- 1.2 **Algorithm:** specifying a set of rules or instructions, or conceiving and representing a procedure that should be adopted or followed by an executor – human, artificial and/or virtual agent – in order realise a task that solves the problem;
- 1.3 **Assessment:** evaluating the quality and suitability of the obtained solution with respect to the original problem.

From this perspective, an individual's cognitive skills can be inferred by observing

him solving a contextualised (computational) task together with other individuals in a given artefactual environment. In particular, the way an individual interacts with other individuals and his contribution to the collective reasoning, respectively their choice of (cognitive) artefacts and his ability to use them correctly and efficiently in the different steps, can be observed directly and used to assess his competence level with respect to the given task. This definition allowed us to define a theoretical framework which explicitly considers the situated nature of CT.

The sound acquisition of complex computational concepts, such as parallelisation and iteration, requires a long-term learning path similar to that followed by individuals to internalise complex mathematical concepts, like counting. Extending Tall's model of mathematical thinking [292], presented in Section 2.3.2 to include CT, we aim to identify the cognitive tools used across three realms in both CT and mathematical thinking. We hypothesise that CT occurs in three worlds of computation, characterised by different types of cognitive and representational artefacts:

- 2.1 **Embodied:** based on embodiment and perception, in which CT is mainly focused on the solution of contextualised problems through ecological and iconic representational cognitive artefacts;
- 2.2 **Symbolic:** based on the conception, description and application of procedures and rules for solving contextualised problems through symbolic (both formal and natural) cognitive artefacts;
- 2.3 **Formal:** based on the creation, generalisation and representation of algorithms through formal languages in order to define structures that can be applied for problem-solving in different, even yet unknown, contexts.

Regarding autonomy, individuals may exhibit varying degrees of involvement in relation to their social interactions and the context of situated cognition. These levels of autonomy are shaped by the collaborative and contextual nature of the activity, where individuals' roles and contributions depend on their interactions with others and the cognitive environment:

- 3.1 **Inactive role:** the individual is not active because he is not able and/or motivated to realise the requested activity.
- 3.3 **Non-autonomous active role:** the individual is motivated for the activity and is able to realise the activity if helped, scaffolded or guided by other members of the situated cognitive system.
- 3.3 **Autonomous active role:** the individual is motivated for the activity and is able to realise autonomously the activity, respectively to guide and/or inspire the other members of the situated cognitive system in the realisation of their activities.

3.3 The CT-cube

We introduce the CT-cube, a three-dimensional framework that provides a comprehensive view of CT and accounts for all three aspects of the above definition. It can be used alongside any CT model for the design, the analysis and the assessment of CT (see Figure 3.1).

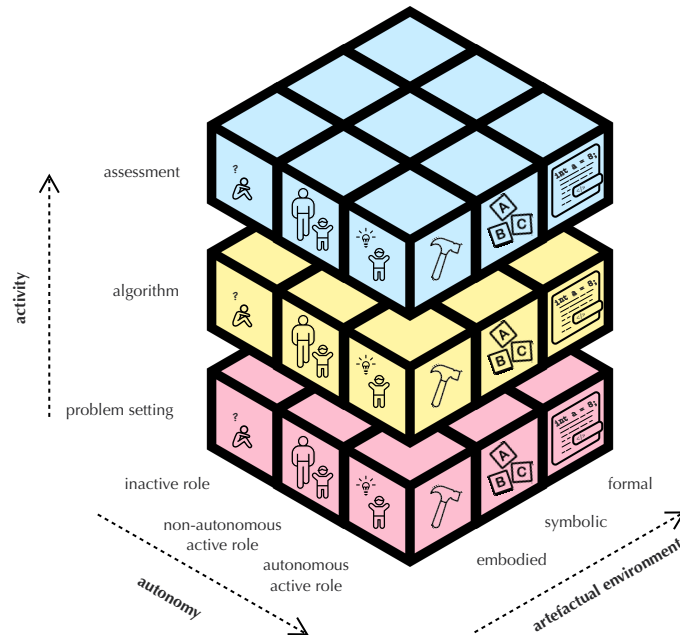


Figure 3.1 – The CT-cube.

This model considers the type of activity (problem setting, algorithm, assessment), the artefactual environment (embodied, symbolic, formal), and the autonomy (inactive role, non-autonomous active role, or autonomous active role).

At each moment of an activity, the situated cognitive system has three dimensions that apply to every individual:

- the type of cognitive **activity** that is being performed or that is required (problem setting, algorithm, assessment);
- the **artefactual environment** or the computational world in which the activities take place, characterised by the tools being used (embodied, symbolic, formal);
- the **autonomy** of the individual with respect to the other individuals in the situated cognitive system (inactive or passive role, active support role or supported individual activity, active leading role or autonomous individual activity).

It is important to remark that the three dimensions are easily observable in practice,

making this framework suitable for the assessment of CT skills.

The CT-cube combines two frames of CT that have been considered extensively in literature: cognitive CT and situated cognitive thinking [150], but while in situated CT, the focus is often concentrated on social and creative skills, the CT-cube adopts a situated cognition view, that consider the whole situated cognitive system, consisting of both the social context and the artefactual environment. In particular, given one or more components of CT, according to a given CT model, a task based on the selected components is designed and realised by explicitly structuring (i) the type of activity that is being performed, (ii) the artefactual environment and (iii) the social interactions and the level of autonomy, a priori and/or during the realisation of the activity.

3.3.1 Framework applications

Design of CT activities The design of CT activities involves structuring the situated cognitive system in a way that promotes effective learning. In an educational context, this includes setting the social environment, organising the artefactual environment, and selecting appropriate problems for students to solve. By strategically guiding these elements, activities can be confined to a subset of the cells within the framework, ensuring a focused learning experience that aligns with desired outcomes. Moreover, the design process can influence the execution of the activities by actively shaping the situated cognitive system. This may involve opening or blocking access to specific cells within the framework, such as adding or removing cognitive artefacts from the environment, thus directing the participants' engagement with particular cognitive and representational tasks.

Assessment of CT activities The assessment of CT activities involves mapping the paths that individuals take within the situated cognitive system during their engagement with the task. This can be achieved through various methods, such as video analysis, to track which cells of the framework are visited over time. By observing the trajectory of these interactions, it is possible to assess individuals' competency levels across multiple dimensions, offering insights into both short- and long-term development of CT skills. This assessment process enables the comparison of individuals across different ages, educational levels, or developmental stages in relation to a specific task. Additionally, it allows for tracking individual progress over time, facilitating a comprehensive evaluation of the progression of CT competencies. The ability to observe these developmental patterns provides valuable feedback for understanding the trajectory of CT learning, from initial exposure to more advanced stages.

Chapter 4

A framework for the analysis and design of Computational Thinking Problems

The content of this chapter has been adapted from the following article with permission of all co-authors and publishers:

- **Adorni, G.**, Piatti, A., Bumbacher, E., Negrini, L., Mondada, F., Assaf, D., Mangili, F., and Gambardella, L. M. (2025). FADE-CTP: A Framework for the Analysis and Design of Educational Computational Thinking Problems. *Technology, Knowledge and Learning* [11].

As an author of this publication, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation, Supervision.

4.1 Summary

This chapter addresses RQ1 by proposing a CT framework for analysing and designing Computational Thinking Problems (CTPs), which are activities that require the application of CT to be solved. By defining CTPs and establishing a general set of core components and characteristics that define these problems, the chapter lays the groundwork for understanding the nature of CT competencies. The characteristics of CTPs are connected to the CT-cube framework (see Chapter 3), particularly focusing on aspects of situated cognition, which highlight the importance of the context and environment in which CT skills are applied.

The chapter discusses how the design and complexity of CTPs impact the skills that students can cultivate, offering insights into designing activities that are both scalable and effective for assessing AT competencies, providing a foundation for RQ2.

Next, a hierarchical structure of CT competencies, which organises them in a way that reflects their interrelated nature. This section supports RQ1 by refining the conceptual framework of CT competencies, which is critical for developing valid assessments.

Finally, the chapter concludes by providing a template for describing the profile of a CTP, which is based on the relationship between the characteristics of a CTP and CT competencies. This template can be adapted to create valid, context-sensitive assessments, thereby contributing to RQ2 and offering guidance for developing tools to assess AT competencies in educational contexts.

4.2 Definition of Computational Thinking Problem

Our definition of CTP is grounded in the CT-cube theoretical framework (see Chapter 3), which integrates foundational concepts of CT from Wing [335] with situated theories of learning by Roth and Jornet [258] and Heersmink [125], framing CT as a dynamic, adaptive process embedded in real-world contexts, rather than a fixed set of competencies. Building on this integrated perspective, we define a CTP as a task that engages learners in applying CT skills to derive solutions within realistic environments, emphasising the influence of physical and social contexts on CT activities and the role of cognitive artefacts in supporting problem-solving. The interplay of the type of activity, the artefactual environment, and the learner's autonomy, key dimensions of the CT-cube, collectively determines the characteristics of the CTP, shaping the competencies developed during engagement and influencing both the learning experience and the effectiveness of problem-solving approaches.

4.2.1 Components

We identified several components that constitute a CTP, illustrated in Figure 4.1.

The *system* comprises the environment and the agent. The *environment* is a physical and/or a virtual external space, characterised by one or more variables, called “descriptors”, which may change over time according to the dynamics of this space. The *agent* is a human, robotic or virtual being that interacts with the environment by performing “actions” to change the value of its descriptors and, therefore, alter the state of the environment. An “algorithm” is a finite set of instructions an agent should follow to perform actions in the environment to solve the task. Algorithms for different agents can take various forms: code for a virtual agent, behaviour for a robot, or a verbal or written set of instructions for a human.

The *problem solver* is a human or group of people who can solve tasks that require the use of algorithms, such as designing, implementing, or communicating them to an

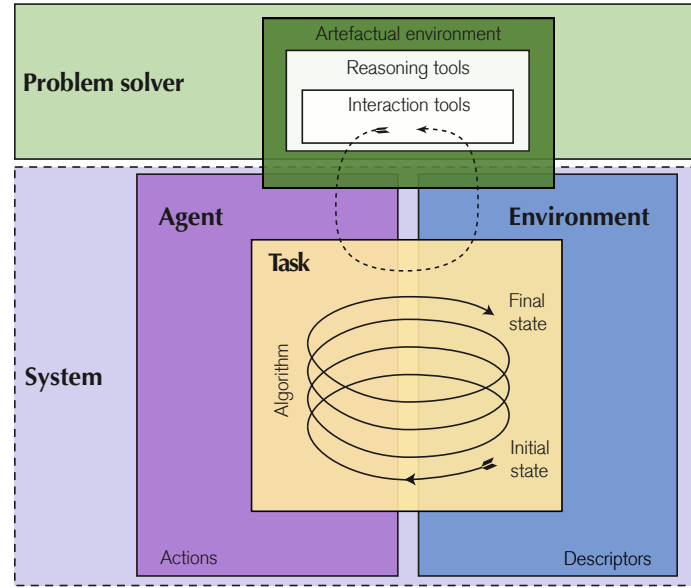


Figure 4.1 – Components of a CTP.

CTPs include (1) the problem solver (in green) characterised by the artefactual environment, i.e., the set of reasoning and interaction tools, (2) the system, which consists of an environment with its descriptors (in blue) and an agent with its actions (in violet), and (3) the task (in yellow) characterised by the set of initial states, algorithms and final states.

agent to change the state of an environment. They have access to *reasoning tools*, which are cognitive artefacts that assist in thinking about the task, such as whiteboards used to organise ideas and understand the logic of a problem or solution. Some of these tools, known as *interaction tools*, also allow the problem solver to interface with the system. For example, a programming platform may serve as both a reasoning tool, enabling the problem solver to plan and design code, and an interaction tool, facilitating the execution of the algorithm and allowing the observation of its effect on the system. Collectively, these tools form the *artefactual environment*, which according to our definition, provided in Chapter 3 and the model of the three worlds of mathematics by Tall [290, 291, 292], can also be categorised in: “embodied”, iconic representational or ecological tools, based on sensory perception and embodiment; “symbolic” tools, used to conceive and apply procedures and rules; and “formal” tools, used to create, generalise and represent structures.

The *task* is the activity that the problem solver performs to find one or more solutions to a CTP. A solution is a combination of “initial states”, “algorithms”, and “final states” that meet the system’s requirements for a particular environment, with its set of states, and a given agent, with its set of algorithms. The initial state is the starting

configuration of the environment, while the final state is the state of the environment after the algorithm is performed. For a solution to be valid, the algorithm must be executed on the initial state and then produce the final state. Each element that composes a task (initial state, algorithm, final states) can be “given” or is “to be found”. Based on the number and the epistemic nature of elements to be found, it is possible to divide tasks into six types. Those with a single objective are: (1) *find the initial state*: given the final state and the algorithm that produced it, the problem solver must infer the initial state on which the algorithm was applied; (2) *find the algorithm*: given the initial and the final states, the problem solver must devise and describe an algorithm, or a part of it, that the agent can execute to transform the system from the initial to the final state; (3) *find the final state*: given the initial state and an algorithm, the problem solver must derive the final state. Pairs of single-objective tasks form those with multiple objectives: (4) *creation act*: a combination of find the algorithm and find the final state; (5) *application act*: a combination find the initial state and find the final state; (6) *project act*: a combination find the initial state and find the algorithm.

4.2.2 Characteristics

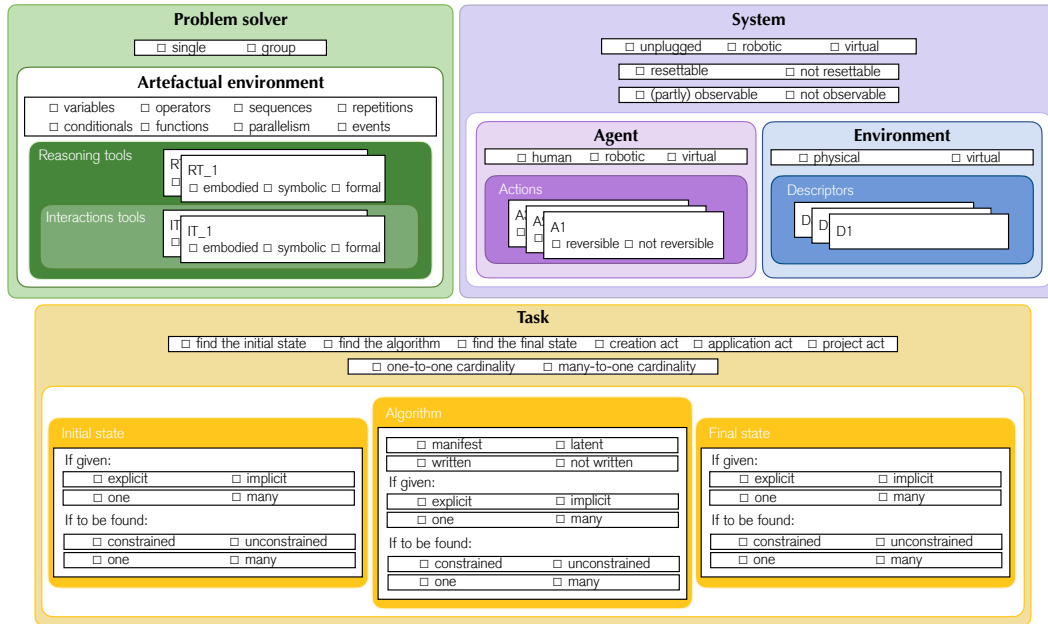


Figure 4.2 – Template for defining components and characteristics of a CTP. The same colour scheme as in Figure 4.1 is applied.

After defining the components of CTPs, we identify key characteristics that further

clarify their nature. These attributes, along with their role, are illustrated in Figure 4.2, which serves as a template for CTP analysis.

Problem domain The category of an activity, determined by the nature of the agent and of the environment.

Three main categories of domains are commonly recognised in cognitive tasks, including: “unplugged” activities, which involve a human agent and a physical environment; “robotic” activities, in which the agent is a robot, and the environment is physical; and “virtual” activities, where both agent and environment are virtual, such as in a simulated scenario.

Tool functionalities The artefactual environment’s capabilities enable the problem solver to construct the algorithm.

The functionalities we included in this categorisation are tailored for beginner-level CT education to introduce foundational algorithmic concepts, such as “variables”, “operators”, “sequences”, “repetitions”, “conditionals”, “functions”, “parallelism” and “events”. For example, a symbolic artefact, such as a block-based programming platform, may have many functionalities, such as sequences, repetitions, conditionals, etc. In contrast, the programming interface may have limited functionalities during a robotic activity, for example, it could only consent using operators (like moving forward) or events.

System resettability The property of a system to be restored to its initial state, either through the direct intervention of the problem solver on the system or indirectly via the reversibility of actions within the system.

Resettability can be “direct” when the problem solver can directly intervene on the system by manually returning the robot to its starting position and restoring the environment; or “indirect” when the problem solver can use a system-provided reset mechanism. If neither option is available, the system is “non-resettable”, for example, when the problem solver can move the robot back to the starting position only through an algorithm, but any environmental alterations remain irreversible.

System observability The property of a system that allows the problem solver to observe the effects of the agent’s actions in the environment and their impact on its state.

Systems can be classified as “totally observable” if every action and their effects are

visible, e.g., if the problem solver and the robot are in the same room, and all changes to the system state are visible in real-time; “partially observable” when only the aggregate effects of a set of actions are visible, e.g., if the problem solver can enter the room only at the end of the task and observe the final state of the system, without seeing the actions that led to it; or “not observable”, if none of the agent’s actions or their results are visible, e.g., if the problem solver cannot enter the room and must infer the system state from other information, such as sensor data. It is worth noting that, in the unplugged domain, problem solver and agent can be the same entity. When they overlap, the system is totally observable.

Task cardinality The relationship between the number of given elements and those to be found to solve a task.

CTPs can present three types of cardinality: “one-to-one”, “many-to-one” or “many-to-many”. In a one-to-one task, each provided element corresponds directly to one element to be found, e.g., if a single initial and a final states are given, a single algorithm has to be found. In a many-to-one task, multiple given elements are intended to be resolved by a single solution element, e.g., if several initial states are provided, and the goal is to find a single algorithm that can transform each of these initial states into the same final state. In a many-to-many task, both the provided and target elements are multiple, requiring the problem solver to find various solutions. For example, a task might provide multiple initial states and a single final state, and the solver would need to identify several algorithms, each capable of transforming one or more of the initial states into the specified final state. This type of task can be traced back to multiple many-to-one tasks.

Task explicitness The level of detail in the presentation of the task’s elements.

In a CTP, the given elements can be “explicit” if they are directly provided and immediately usable in the problem-solving process, or “implicit” if they are expressed with constraints that require further interpretation to be understood. For example, in a task where the problem solver must find the algorithm for a robot to turn on its lights after finding a ball, the ball’s position can be given explicitly (e.g., coordinates) or implicitly (e.g., in the playground).

Task constraints The limitations or specific requirements that the task elements to be found must meet to consider the solution valid.

In a CTP, the elements to be found can be “unconstrained” if they can be freely selected

among all possible states and algorithms, with no limitations or specific requirements that need to be met to consider the solution valid; or “constrained” if they must belong to a restricted subset of states or algorithms. Referring to the same example presented to explain the task explicitness characteristic, the algorithm to be found can be unconstrained if the robot can perform any action to find the ball (e.g., moving randomly, using sensors, etc.) or constrained if the programming platform limits the robot’s actions (e.g., restrict movement to specific directions, using only specific sensors).

Algorithm representation The mean by which an algorithm is given.

An algorithm is considered “manifest” if directly expressed, while “latent” if not stated but should be inferred by the problem solver. Manifest algorithms can be “written” if represented by an external and persistently, like the code in a programming language, or “not written” if communicated verbally or through other non-permanent means.

4.3 Catalogue of Computational Thinking competencies

Alongside the definition of CTPs, their components, and their characteristics, we have developed a catalogue of CT competencies that are fundamental abilities students need to solve CTPs effectively. This integration is primarily theoretical, synthesising existing competencies frameworks to propose a structured perspective on the relationship between CT and CTPs. To ensure a through approach, we drew from multiple state-of-the-art competency models and frameworks. Our selection was informed by the literature reviews of Tikva and Tambouris [299] and Bocconi et al. [37, 38], which provide a comprehensive overview of CT skills in compulsory education.

A key framework that guided the development of our catalogue is that of Brennan and Resnick [43], which categorises CT skills into computational concepts, practices, and perspectives. While this model is commonly referenced in literature, it primarily focuses on activities conducted in digital environments, such as programming and software development. Although the CT aspects covered are essential, they do not encompass the broader range of CTPs we explore, including hands-on robotics and unplugged activities. To bridge this gap, we extended the framework to incorporate competencies applicable across these varied contexts, such as elements from the STEM taxonomy proposed by Weintrop et al. [326], which includes data practices, modelling and simulation practices, computational problem-solving, and systems thinking practices. We also referenced the work of Shute et al. [275], which expands on the previous frameworks by offering a broader, more adaptable set of CT competencies with a focus on cognitive processes and applicability across diverse contexts.

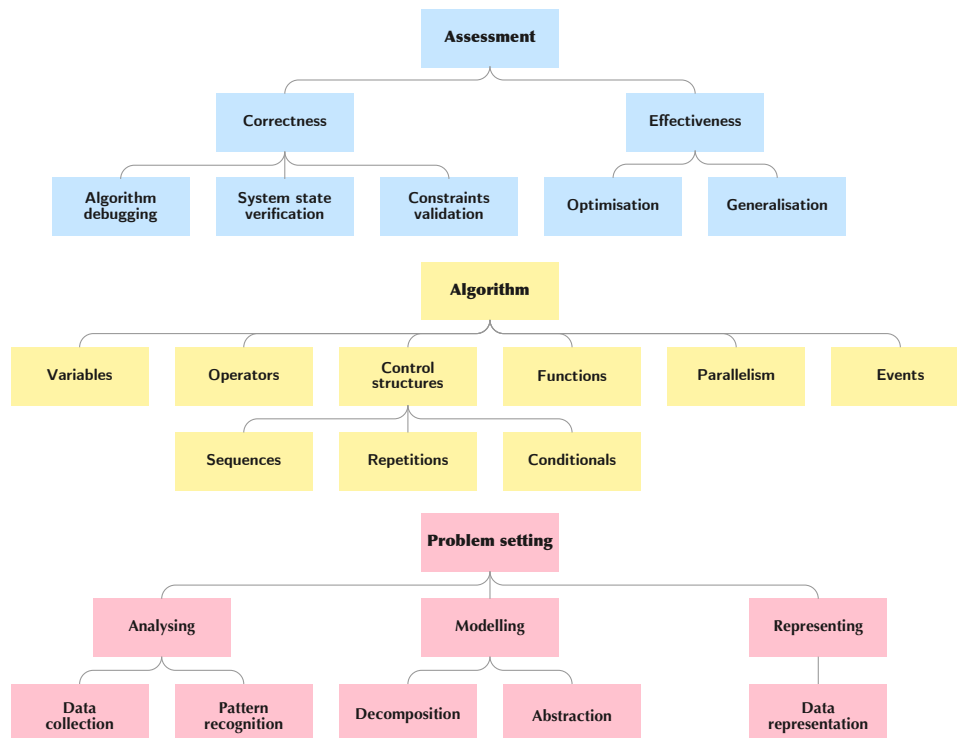


Figure 4.3 – Taxonomy of CT competencies.

The overall structure is based on CT-cube framework. The sub-skills are derived from validated CT models by Brennan and Resnick [43], Shute et al. [275], Weintrop et al. [326]. The same colour scheme as in Figure 3.1 is applied.

To provide a comprehensive framework for CT competencies, we organised our catalogue into a hierarchy of skills and sub-skills (see Figure 4.3). This structure clarifies the relationships among competencies, making it easier to identify specific skills within broader categories and supporting a more precise and targeted approach for educators and researchers working with CT skill development and assessment. The first layer of competencies (see Table 4.1) is based on the activity dimension of the CT-cube, while the lower layers (see Tables 4.2 to 4.4) are based on the frameworks of Brennan and Resnick [43], Shute et al. [275], Weintrop et al. [326]. While this framework provides a structured theoretical foundation, further empirical validation is necessary to confirm the relationships proposed.

Table 4.1 – Core skills definition (level 1).

Competence (level 1)	Definition
<i>Problem setting</i> ^a	Recognise, understand, reformulate or model a CTP and its components so that its solution can be computed.
<i>Algorithm</i> ^b	Conceive and represent a set of agent’s actions that should be executed by a human, artificial or virtual agent to solve the task.
<i>Assessment</i> ^c	Evaluate the quality and validity of the solution in relation to the original task.

These skills are based on the values of the activity dimension of our CT-cube framework.

^a See Table 4.2 for “problem setting” sub-competencies.

^b See Table 4.3 for “algorithm” sub-competencies.

^c See Table 4.4 for “assessment” sub-competencies.

Table 4.2 – Problem setting sub-skills definition (level 2 and 3).

Competence (level 2)	Competence (level 3)	Definition
<i>Analysing</i>		Collect, examine and interpret data about the system: environment descriptors and agent actions.
	<i>Data collection</i>	Gather details about the system.
	<i>Pattern recognition</i>	Identify similarities, trends, ideas and structures within the system.
<i>Modelling</i>		Restructure, clean and update knowledge about the system.
	<i>Decomposition</i>	Divide the original task into sub-tasks that are easier to be solved.
	<i>Abstraction</i>	Simplify the original task, focus on key concepts and omit unimportant ones.
<i>Representing</i>		Illustrate or communicate information about system and task.

The skills listed are based on leading-edge competence models (Angeli et al. [19], Barr and Stephenson [25], Bocconi et al. [37], Brennan and Resnick [43], Csizmadia et al. [67], Selby [271], Selby and Woollard [272], Shute et al. [275], Thalheim [295], Weintrop et al. [326], Wing [337]).

Table 4.3 – Algorithm sub-skills definition (level 2 and 3).

Competence (level 2)	Competence (level 3)	Definition
<i>Variables</i>		Entity that stores values about the system or intermediate data.
<i>Operators</i>		Mathematical operators (such as addition (+), subtraction (−) etc.), logical symbols (such as and (&), or (), and not (!)) or for comparison (such as equal to (==), greater than (>), and less than (<)), or even specific commands or actions (such as “turn left” or “go straight”).
<i>Control structures</i>		Statements that define the agent actions flow’s direction, such as sequential, repetitive, or conditional.
	<i>Sequences</i>	Linear succession of agent actions.
	<i>Repetitions</i>	Iterative agent actions.
	<i>Conditionals</i>	Agent actions dependent on conditions.
<i>Functions</i>		Set of reusable agent actions producing a result for a specific sub-task.
<i>Parallelism</i>		Simultaneous agent actions.
<i>Events</i>		Variations in the environment descriptors that trigger the execution of agent actions.

The skills listed are based on leading-edge competence models (Bocconi et al. [37, 38], Brennan and Resnick [43], Cui and Ng [68], Rodríguez-Martínez et al. [250], Shute et al. [275]).

Table 4.4 – Assessment sub-skills definition (level 2 and 3).

Competence (level 2)	Competence (level 3)	Definition
<i>Correctness</i>		Assess whether the task solution is correct.
	<i>Algorithm debugging</i>	Evaluate whether the algorithm is correct, identifying errors and fixing bugs that prevent it from functioning correctly.
	<i>System states verification</i>	Evaluate whether the system is in the expected state, detecting and solving potential issues.
	<i>Constraints validation</i>	Evaluate whether the solution satisfies the constraints established for the system and the algorithm, looking for and correcting eventual problems.
<i>Effectiveness</i>		Assess how effective is the task solution.
	<i>Optimisations</i>	Evaluate whether the solution meets the standards in a timely and resource-efficient manner, and eventually identify ways to optimise the performance.
	<i>Generalisation</i>	Formulate the task solution in such a way that it can be reused or applied to different situations.

The skills listed are based on leading-edge competence models (Bocconi et al. [37], Brennan and Resnick [43], Shute et al. [275], Weintrop et al. [326]).

4.4 Profiling Computational Thinking Problems

This section introduces our framework for profiling CTPs, building on our earlier discussions of CTP components and characteristics. We defined specific relationships between CTP characteristics and CT competencies, outlining, for each skill, the set of characteristics essential for their development and those that inhibit it. All required characteristics must be present to develop a specific competence, and none may be inhibitory.

While identifying the required and absent characteristics allows us to determine which competencies can technically be developed, this perspective is somewhat limited. Therefore, we also included characteristics that can enhance and support skill development beyond basic requirements, strengthening the overall framework for competency development. For example, a manifest written algorithm can significantly facilitate the development of algorithmic skills at different levels of abstraction, such as repetitions, by helping learners understand how loops work, recognise them and practice their application, ultimately leading to assimilation [36, 106].

The relationships between CTP characteristics and CT competencies are examined in detail in Appendix B, where we outline (i) how various CTP characteristics influence the development of CT competencies and (ii) which CT skills are more frequently developed and/or employed when solving CTPs with specific traits. While this integration is theoretically informed by existing research, further empirical validation is necessary to refine these relationships.

Table 4.5 illustrates our framework and serves as a template for analysing and designing CTPs by creating a profile of each specific CTP.

Table 4.5 – Profiling template for CTPs.

Rows represent the CT competencies in our catalogue and columns the CTPs characteristics, including tools' functionalities, system's property and task traits. The same colour scheme as in Figures 3.1 and 4.1 is applied.

		Tool functionalities								System				Task									
		Variables	Operators	Sequences	Repetitions	Conditionals	Functions	Parallelism	Events	System resettable	System not resettable	System (partially) observable	System not observable	One-to-one cardinality	Many-to-one cardinality	Explicit elements	Implicit elements	Unconstrained elements	Constrained elements	Algorithm manifest	Algorithm latent	Algorithm written	Algorithm not written
Problem setting	Data collection	✓							+	+	+		+	+	+	+	+	+	+	+	+	+	+
	Pattern recognition	+		+	✓*	+		✓*	+	+	+		+	+	+	+	+	+	+	+	+	+	+
	Decomposition	+	+	✓*	+	+		✓*	+	+	+		+	+	+	+	+	+	+	+	+	+	+
	Abstraction	✓		+		+		✓		+		+		+	+	+	+	+	+	+	+	+	+
	Data representation	✓		+	+	+	+	+			+		+		+		+		+	+	+	+	+
Algorithm	Variables	✓	+	+	+	+	+	+	+			+	+	+	+	+	+	+	+			+	+
	Operators	+	✓	+	+	+	+	+	+			+	+	+	+							+	+
	Sequences	+	+	✓	+		+					+	+		+	+	+	+	+			+	+
	Repetitions	+	+	+	✓			+				+	+	+	+	+	+	+	+			+	+
	Conditionals	+	+			✓			+			+	+	+	+	+	+	+	+			+	+
	Functions	+	+	+	+		✓					+		+	+	+	+	+	+			+	+
	Parallelism	+	+					✓				+		+	+					+		+	+
	Events	+	+			+			✓			+		+	+	+	+	+	+			+	+
Assessment	Algorithm debugging	+	+	+	+	+	+	+	+	✓	✗	+		✓				+		✗	✓	✗	✗
	System state verification									✓	✗	+				+			✓	✗	✓	✗	✗
	Constraints validation	+	+	+	+	+	+	+	+	✓	✗	+				✗	✓						
	Optimisation	+	+	+	+	+	+	+	+	✓	✗	+											
	Generalisation	✓		+	+	+	✓	+	+	✓	✗	+		+		+		+					

✓ indicates that the characteristics is required for the development of the competence.

✓* indicates that at least one of several characteristics in a group is required for the development of the competence.

✗ indicates that the characteristic prevents the development of the competence.

+

Blank cells indicate that the characteristic is irrelevant to the development of the competence.

4.4.1 Framework applications

Analysis of CTPs The analysis of CTPs involves identifying the characteristics that define a task, enabling the determination of which competencies can be developed or assessed based on the presence of necessary traits and the absence of inhibitory ones. By evaluating the structural and contextual elements of a CTP, this methodology provides a detailed understanding of how different tasks align with specific CT competencies, aiding both in assessment and identifying areas for improvement.

Design of CTPs The design of CTPs involves selecting the specific competencies to be targeted for development and/or assessment. Once the desired competencies are identified, it is essential to determine the necessary characteristics to include in the task, ensuring that any elements that could inhibit skill development are excluded. Furthermore, supportive characteristics that may not be strictly required but could enhance the learning process can be selectively integrated, enriching the CTP and making it more effective in fostering the intended competencies. This approach ensures that the design of CTPs is purposeful, targeted, and conducive to achieving the desired outcomes.

Chapter 5

Probabilistic modelling for Intelligent Assessment Systems

The content of this chapter has been adapted from the following articles with permission of all co-authors and publishers:

- Antonucci, A., Mangili, F., Bonesana, C., and **Adorni, G.** (2022). Intelligent Tutoring Systems by Bayesian Nets with Noisy Gates. *The International FLAIRS Conference Proceedings* [20].
- Mangili, F., **Adorni, G.**, Piatti, A., Bonesana, C., and Antonucci, A. (2022). Modelling Assessment Rubrics through Bayesian Networks: a Pragmatic Approach. In *2022 International Conference on Software, Telecommunications and Computer Networks* (SoftCOM) [181].
- **Adorni, G.**, Mangili, F., Piatti, A., Bonesana, C., and Antonucci, A. (2023a). Rubric-based Learner Modelling via Noisy Gates Bayesian Networks for Computational Thinking Skills Assessment. *Journal of Communications Software and Systems* [9].

As an author of these publications, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation.

5.1 Summary

This chapter explores probabilistic models, particularly BNs, for developing IAS, addressing RQ3. It outlines how BNs with noisy gates can model student skills, comparing these models to Bayesian Knowledge Tracing (BKT). The discussion provides insights into the integration of probabilistic methods into IAS, which can help improve the precision and adaptability of assessments across various educational contexts.

Furthermore, the chapter examines how assessment rubrics can be modelled with BNs to evaluate competencies, including the ordering of skills and the inclusion of supplementary competencies to offer a more comprehensive assessment of student performance. This directly supports RQ3 by presenting a probabilistic framework for assessing AT competencies in diverse contexts, enhancing the ability to model complex relationships between skills.

5.2 Bayesian Networks and learner models

BNs can be employed to represent learner skills and observable actions in the context of IASs. Key notations used include uppercase letters to denote variables (e.g., X , Y), lowercase letters for states (e.g., y_E), bold letters for sets of variables (e.g., \mathbf{X} , \mathbf{Y}).

The structure of a BN over a set of variables is described by a directed acyclic graph \mathcal{G} whose nodes are in one-to-one correspondence with the variables in the set. We call parents of a variable X , according to \mathcal{G} , all the variables are connected directly with X with an arc pointing to it. Learner models usually include a set of n latent (i.e., hidden) variables $\mathbf{X} := (X_1, \dots, X_n)$, henceforward referred to as *skill nodes*, describing the competence profile of the learner and some m manifest variables $\mathbf{Y} := (Y_1, \dots, Y_m)$, hereafter called *answer nodes*, describing the observable actions implemented by the learner to answer each specific task.

While the orientation of a BN arc may not necessarily reflect a causal interpretation, in practice, graphs that implement an IAS often have a bipartite structure that includes arcs from the skills to the questions but not vice versa. This means that each question receives incoming arcs from the relevant skills for answering the question. By adopting this bipartite structure, we can model assessment rubrics more suitably. This results in a set of simple and interpretable relations that model how the presence or absence of a specific competence directly affects the learner's behaviour when solving tasks that require such competence. For this purpose, we only consider the case of binary skill nodes that take the value of 1 or the "true" state, indicating whether the pupil possesses the skill. Additionally, we use binary answer nodes that denote a correct answer or determine whether the pupil has shown the desired behaviour when solving the task.

The relations of a BN-based learner model (skills and questions) can be graphically depicted as in the example of Figure 5.1. The answer nodes describe whether the learner has been able or not to program, for example, a maze game (Y_1) or a statistical model (Y_2). The skill nodes represent the ability to build this program using a block-based programming language such as Scratch (X_1) or a text-based programming language such as Python (X_2). The second skill can be applied to answer both ques-

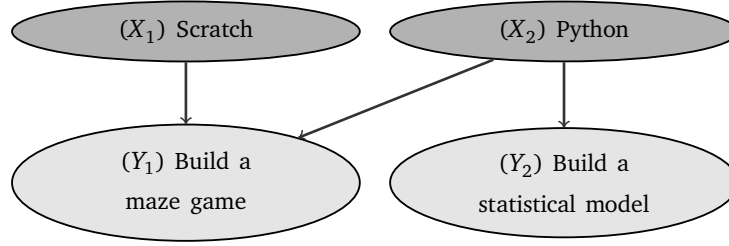


Figure 5.1 – Example of BN-based learner model.

tions and therefore, X_2 is a parent node for both answer nodes Y_1 and Y_2 . Instead, the first skill can be used to answer just the first question, and therefore, there is no direct arc from X_1 to Y_2 .

Once the graph \mathcal{G} structuring the BN is established, the definition of the BN over the $n + m$ variables of the network $\mathbf{V} := (V_1, V_2, \dots, V_{n+m})$, including both skills (\mathbf{X}) and answers (\mathbf{Y}), consists in a collection of Conditional Probability Tables (CPTs) giving the probabilities $P(Y_i = 1 | \text{Pa}(Y_i))$ that Y_i takes value one given all possible joint states of its parent nodes $\text{Pa}(Y_i)$. Let \mathbf{V} take values in $\Omega_{\mathbf{V}}$, the independence relations imposed from \mathcal{G} by the *Markov condition*, i.e., the fact that each node is assumed to be independent of its non-descendants non-parents given its parents, induce a joint probability mass function over the BN variables that factorises as follows [161]:

$$P(\mathbf{V} = \mathbf{v}) = \prod_{v \in \mathbf{V}} P(v | \text{pa}(V)), \quad (5.1)$$

where $\mathbf{v} = (v_1, v_2, \dots, v_{n+m})$ represents a given joint state of the variables in \mathbf{V} . BN inference consists of the computation of queries based on Equation (5.1). In particular, we are interested in *updating* tasks consisting in the computation of the marginal posterior probability mass function for a single skill node $X_q \in \mathbf{X}$ given the observed state \mathbf{y}_E of the answer nodes $\mathbf{Y}_E \subseteq \mathbf{Y}$:

$$P(x_q | \mathbf{y}_E) = \frac{\sum_{\mathbf{v} \in \Omega_{\mathbf{V}} \setminus (x_q, \mathbf{y}_E)} \prod_{v \in \mathbf{V}} P(v | \text{pa}(V))}{\sum_{\mathbf{v} \in \Omega_{\mathbf{V}} | \mathbf{y}_E} \prod_{v \in \mathbf{V}} P(v | \text{pa}(V))}, \quad (5.2)$$

where $\Omega_{\mathbf{V} | \mathbf{v}'} := \{\mathbf{v} : v_i = v'_i \ \forall \ v'_i \in \mathbf{v}'\}$.

According to the above model, multiple parent skills may be relevant to the same answer. The challenges in the existing model are primarily twofold. First, the elicitation process involves an exponential number of parameters due to the potential involvement of multiple parent skills in determining a single answer. Assuming that the answer node Y_j has n parent skills, this results in 2^n parameters to be elicited by experts. This high number of parameters to be elicited by experts might discourage practitioners from

using these tools in their applications because of a too-demanding elicitation process when many skills are affecting the answer to a question. Second, the inference task in Bayesian networks, as described in Equation (5.2), is NP-hard in the general case and highly dependent on the complex graph topology, which is in practice exponential in the graph tree-width [161], making it computationally demanding, especially for models with a high number of parent nodes (high maximum *indegree*), as this involves both large CPTs and tree-width.

To address these issues, in the next section, we propose a solution that introduces noisy gates, specifically the noisy-OR gate [220], to reduce the number of parameters required for model elicitation, shifting from an exponential to a linear relationship with the number of relevant skills. This reduction in parameters streamlines the expert elicitation process and mitigates the potential discouragement of IAs practitioners. Additionally, noisy gates enhance the efficiency of inference tasks, enabling faster computations. Furthermore, the model recognises that a disjunctive relation among skills may not always be applicable in practice. To accommodate diverse scenarios, using more general logical functions, such as conjunctive relations, is considered, providing flexibility and realism to the learner model while benefiting from reduced parameter complexity in elicitation and inference.

5.3 Noisy gates

5.3.1 Disjunctive gates

The noisy-OR network induces the following CPT between the n parent skill nodes $\mathbf{X} = (X_1, \dots, X_n)$ and the observable answer node Y_j [220]:

$$P(Y_j = 0 | \mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)) = \prod_{i=1}^n (\mathbb{I}_{\mathbf{x}_i=0} + \lambda_i \mathbb{I}_{\mathbf{x}_i=1}), \quad (5.3)$$

where \mathbb{I}_A is the indicator function returning one if A is true and zero otherwise. The second term $\lambda_i \mathbb{I}_{\mathbf{x}_i=1}$ represents the noise as it introduces the possibility that a skill X_i that the student possesses is not expressed in task Y_j (this phenomenon is also called *slip* elsewhere in this work). The value of λ_i implying the biggest uncertainty associated with the task-skill pair (Y_j, X_i) is 0.5, whereas the value $\lambda_i = 0$ models the certainty that skill X_i , whenever present, will be expressed in solving task Y_i and, vice versa, $\lambda_i = 1$ model the fact that X_i cannot be expressed in task Y_j .

Figure 5.2 shows a typical representation of the structure of the noisy-OR network, introducing n auxiliary variables (also called inhibitor nodes), which help clarify Equation (5.3). To reduce the number of parameters, the structure of this network defines

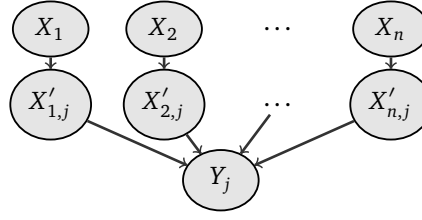


Figure 5.2 – A noisy gate explicit formulation.

deterministically the state of Y_j as the logical disjunction (OR) of the auxiliary parent nodes. This first simplification removes the need to specify the answer node CPT given the state of its parent nodes. Furthermore, the noisy-OR structure sets the input variable X_i as the unique parent of $X'_{i,j}$ and constraints $X'_{i,j}$ to be zero with probability one when $X_i = 0$. The relationship between skill and answers would be purely logical-deterministic were it not for the noise introduced by the so-called inhibition parameters $\lambda_{i,j} = P(X'_{i,j} = 0 | X_i = 1)$, representing the probability of not expressing skill i in task j . This is, thus, the only parameter to be determined.

Auxiliary variables can be interpreted as *inhibitors* of the corresponding skills. We can regard the auxiliary variable $X'_{i,j}$ as an inhibitor of skill X_i in performing the action described by Y_j , since with probability $\lambda_{i,j}$ it makes the skill unavailable to the success of Y_j even if the skill X_i is indeed mastered by the learner. It can be regarded as analogous to the slip probability in BKT models. In accordance with the above description of the noisy-OR gate, missing skill i implies the inability to apply it to any question j , whereas if the learner has the skill, the probability of being able to apply it depends on the specific task and is equal to $1 - \lambda_{i,j}$. The model parameters should, therefore, be related in some sense to the task's difficulty. For instance, they can be in the false state, e.g., $X'_{i,j} = 0$ (with probability $\lambda_{i,j}$), even when the corresponding skill node X_i is in the true state, indicating that, although the learner possesses the skill, it cannot be expressed in task Y_j . By defining the probability of a failure in expressing a possessed skill in the specific task j , the inhibition parameter $\lambda_{i,j}$ provides a measure of the task difficulty. If a pair skill-answer has a large inhibition, the state of the answer node tells, in general, little about the state of the skill node. The extreme case of $\lambda_{i,j} = 1$ corresponds to a missing arc in the BN graph between skill i and answer j .

5.3.2 Leaky models

In a noisy-OR gate, when all skills are missing, all auxiliary variables are false; therefore, all answers must be wrong. Such a model excludes the possibility of a lucky guess. To avoid this, the noisy gates are made *leaky* by adding a leak node, which represents the possibility of a random guess, i.e., a correct answer or a behaviour given without

mastering any required competencies. The leak is a boolean variable playing the role of an auxiliary skill node X_{leak} , which is set in the observed state $X_{j,\text{leak}} = 1$, and added to the parents of all answer nodes for which random guessing is possible. The chances of guessing answer Y_j at random, i.e., without mastering any of the relevant competencies, is given by parameter $1 - \lambda_{j,\text{leak}}$. For instance, in a multiple choice question with four options, one of which is correct, one should set $1 - \lambda_{j,\text{leak}} = \frac{1}{4}$. $1 - \lambda_{j,\text{leak}}$ can therefore be seen as the analogous of the guess probability in BKT [59].

To apply the above model, the domain expert (e.g., the teacher) should first list the parent-less skill nodes (including, eventually, the leak) X_1, \dots, X_n , the childless answer nodes Y_1, \dots, Y_m and connect by an arc the skills to all answer nodes in which they can be used. Then, the instructor should quantify for each pair of skill-answer nodes, X_i and Y_j , connected by an arc, the value of the inhibition $\lambda_{i,j}$. This results in a total of at most $n \cdot m$ parameters to be elicited. Finally, the expert should state each skill's prior probabilities π_i .

5.3.3 Comparison with Bayesian Knowledge Tracing

While the BKT, in its standard implementation, traces the evolution of a single skill over time, our approach focuses on fine-grained skills modelling at the specific moment the assessment is performed. However, a parallel can be drawn between the two. BKT models student knowledge at time t as the (binary) latent variable $X(t)$ of a Hidden Markov Model (HMM) [59]. Learning is modelled as the transition of $X(t)$ from state zero (lack of knowledge) to state one (knowledge acquired). The model defines four parameters: (i) the *initial* probability, i.e., the probability that the knowledge has been already acquired at the beginning of the activity; (ii) the *learning* probability, that is, the probability of acquiring the probability between t and $t + 1$; (iii) the *slip* probability of making a mistake when the knowledge is acquired; (iv) the *guess* probability of doing right in the lack of knowledge.

In our model, the probability of the *slip* may vary depending on the pair skill i and task j , represented by the inhibition $\lambda_{i,j}$. The *guess* probability depends on the task and is equal to $1 - \lambda_{\text{leak},j}$. The *initial* probability of a skill X_i is defined by its prior probability π_i . Notice, however, that since our approach, differently from BKT, does not model the learning process, the concept of initial probability here is meant to describe our initial knowledge of the learner competence profile rather than the probability that the skill is initially acquired. For the same reason, no *learning* probability is defined in our model.

5.4 Assessment rubrics

Several possible approaches exist to identify the knowledge components to be included in a learner model. We decided to consider only assessment methods based on a task-specific assessment rubric for assessing a given competence through a given task or family of similar tasks [50, 148].

A task-specific assessment rubric consists of a two-entry table where each row corresponds to a component of the given competence, described in the light of the given task. In contrast, each column corresponds to a competence level in ascending order of proficiency. For each combination of component and level, the rubric provides a qualitative description of the behaviour expected from a person with the given level in the given component. Identifying a person's competence level consists of matching the learner's behaviours while solving a given task with those described in the assessment rubric.

For instance, Table 5.1 shows the task-specific assessment rubric for an example focused on assessing the student's ability to use iterative instructions in algorithms. This competence has two levels depending on the tools used by the learner: a visual programming language (X_1) or a textual programming language (X_2). By checking how the learner produced the algorithm, the teacher can see whether he applied any of the methods in the rubrics and assign him the corresponding competence level.

Table 5.1 – Example of a task-specific assessment rubric.

In this rubric, there is only one competence component, the ability to design an algorithm containing loops, and two competence levels, the ability to do it using either a block-based programming language or a text-based programming language.

		Competence level	
		X_1 $c = 1$	X_2 $c = 2$
Competence component	Loops $r = 1$	Develop an iterative algorithm using a block-based programming language	Develop an iterative algorithm using a text-based programming language

In assessment rubrics, the ordering between competence levels, and sometimes between competence components, plays a fundamental role. A competence level or component is considered higher than another if the former implies the latter, meaning that a learner with the higher competence can also perform all the tasks that require the lower. In practice, the competence level matching the student's behaviours for a given

component does not always correspond to the actual learner's state of knowledge. It is also possible that the person possessed a higher level but is underperforming.

To ensure completeness, rubrics must include a basic competence level, describing the observed behaviours of learners who have not yet acquired the competence in question. This level is typically expressed in a constructive manner, highlighting what the learner is capable of rather than focusing on what they cannot do. When evaluating a person using a rubric, the final assessment must correspond to one of the rubric's defined levels.

In the case of a task composed of similar sub-tasks., i.e., tasks sharing the same assessment rubric, it is possible to observe behaviours corresponding to different competence levels across various sub-tasks. As a result, the competence level identified in a given instance does not necessarily reflect the learner's overall state of knowledge, as external factors or temporary difficulties may lead to underperformance.

In the following subsection, we illustrate how this uncertainty can be considered and how an overall assessment based on a full battery of tasks can be produced by modelling the learner competence profile with the BN-based approach described in Section 5.2.

5.5 Modelling assessment rubrics by Bayesian Networks

Considering a task-specific assessment rubric, as defined above, it is possible to derive a learner model, as presented in Section 5.2, hereafter referred to as *baseline model*. For each cell (c, r) of an assessment rubric with R rows and C columns, we introduce a latent binary competence variable X_{rc} , taking value one for a learner mastering the corresponding competence level and zero otherwise. Moreover, for each task t , in a battery of T similar tasks, and each competence variable X_{rc} , we define an observable (manifest) binary variable Y_{rc}^t taking value one if the behaviour described in the assessment rubric's cell (r, c) was applied successfully by the learner in solving task t and zero if he failed using it.

In addition, we extend this baseline model in two ways. Firstly, we explicitly impose the *ordering of competence levels* encoded by the rubric. Secondly, we include in the model task-specific *supplementary skills*, which can be combined with each other and with the competencies of the rubric through arbitrary logic functions.

5.5.1 Ordering of competences

In the baseline model, as described in Section 5.2, it was indirectly accounted for the partial ordering between variables by setting as parents of answer node Y_{rc}^t the skill

node X_{rc} and all skill nodes corresponding to higher competence levels. The network was quantified through noisy-OR relations, as described in Section 5.3.1. This structure assumes that an observed behaviour can be explained as the student mastering the corresponding competence level or a higher one if he is underperforming, thus not exploiting his full potential, but cannot be achieved through a lower level.

As mentioned above, we interpret the (partial) ordering between competencies defined by the assessment rubric as implication constraints, meaning that possessing a particular skill X_i implies that the learner also possesses his inferior competencies. While exploited to design the network structure, this hierarchy of competencies is not strictly imposed by the above baseline model, giving rise to posterior inferences that are usually inconsistent.

To solve this issue, we enrich the model by adding an auxiliary variable D_{ik} for each relation $X_i \implies X_k$ defined by the rubric. A constraint node D_{ik} is always in the observed state one, with X_i and X_k as parent nodes. The desired implication constraint is then implemented by choosing a CPT for D_{ik} such that $P(D_{ik} = 1 | X_i = 1, X_k = 0) = 0$. The addition to the network of each observed node D_{ik} changes the prior probabilities of X_i and X_k , initially set to π_i and π_k . Let

$$\begin{aligned} p_{00} &= P(D_{ik} = 1 | X_i = 0, X_k = 0) \\ p_{01} &= P(D_{ik} = 1 | X_i = 0, X_k = 1) \\ p_{11} &= P(D_{ik} = 1 | X_i = 1, X_k = 1), \end{aligned} \quad (5.4)$$

be the non-null parameters in the CPT of D_{ik} . After updating with the evidence $D_{ik} = 1$, one has

$$\begin{aligned} P(X_i = 1 | D_{ik} = 1) &= \frac{p_{11} \pi_j \pi_k}{K}, \\ P(X_k = 1 | D_{ik} = 1) &= \frac{p_{11} + p_{01} \pi_j \pi_k}{K}, \end{aligned} \quad (5.5)$$

with $K = p_{11} \pi_j \pi_k + p_{01} (1 - \pi_j) \pi_k + p_{00} (1 - \pi_j) (1 - \pi_k)$.

In this work, we simply assume $p_{00} = p_{01} = p_{11}$ and adopt uniform prior probabilities $\pi_i = \pi_k = 0.5$. Applying them to Equation (5.5) give $P(X_i = 1) = \frac{1}{3}$ and $P(X_k = 1) = \frac{2}{3}$. This result follows from the fact that skill X_i can only be possessed jointly with X_k , whereas X_k can also be owned when $X_i = 0$.

Under the assumption $p_{00} = p_{01} = p_{11} = p_*$, the prior over the superior skill X_i can be interpreted as the conditional probability of having it given that the learner possesses the inferior skill X_k since

$$P(X_i = 1 | X_k = 1, D_{ik} = 1) = \frac{\pi_i \pi_j p_*}{\pi_i \pi_j p_* + (1 - \pi_i) \pi_j p_*} = \pi_i. \quad (5.6)$$

5.5.2 Supplementary competencies

While the assessment rubric details the components of the competence of interest and their interactions with the specific task and available tools, it does not necessarily include all the skills required to solve the task successfully.

For instance, considering the assessment rubric proposed in Table 5.1, to develop an iterative algorithm with a text-based programming language successfully, the learner might also need knowledge about the different types of statements, e.g., while, repeat, for, do until and so on. Ignoring such supplementary skills might be misleading in an automatic assessment system, as failures due to the lack of one of them would not be recognised as such and, eventually, be attributed to the absence of the competence components under assessment. Therefore, if not adequately modelled, the lack of unmodelled supplementary skills would translate into an unfairly negative evaluation of the competencies of interest.

To produce fairer assessments, we extend the model by an additional layer of auxiliary nodes combined with a logic function to allow for the inclusion of a suitable set of supplementary skills.

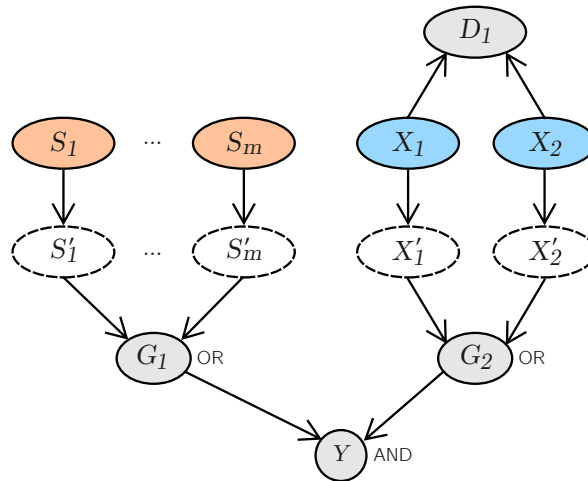


Figure 5.3 – Example of BN modelling a task-specific assessment rubric.

The rubric has two cells, represented by skills X_1 and X_2 (on the right, in light blue), m supplementary skills grouped in a single set (on the left, in orange), and the constraint $X_2 \implies X_1$, represented by the auxiliary variable D_1 (on the top right, in light grey).

Figure 5.3 shows an example of the structure of the extended network. Supplementary skills are described by additional skill nodes S_1, \dots, S_m , which are grouped into sets of interchangeable skills (in the case of the example, we have just one set). Each of these groups is connected through a noisy-OR to a node in the layer of auxiliary

latent nodes, hereafter referred to as group nodes G_1, \dots, G_l , representing the success or failure in applying the type of competence described by each group to the specific task Y . Finally, the group nodes are connected to the answer node through a logic AND or any other logic function suitable for the particular task.

When supplementary skills can be directly assessed through observing specific learner behaviours or by purposed questions, additional answer nodes can be added to the network as direct children of the relevant supplementary skills.

Part III

Assessment instruments

Chapter 6

The unplugged Cross Array Task

The content of this chapter has been adapted from the following article with permission of all co-authors and publishers:

- Piatti, A., **Adorni, G.**, El-Hamamsy, L., Negrini, L., Assaf, D., Gambardella, L., and Mondada, F. (2022). The CT-cube: A framework for the design and the assessment of computational thinking activities. *Computers in Human Behavior Reports* [227].
- **Adorni, G.**, Piatti, A., Bumbacher, E., Negrini, L., Mondada, F., Assaf, D., Mangili, F., and Gambardella, L. M. (2025). FADE-CTP: A Framework for the Analysis and Design of Educational Computational Thinking Problems. *Technology, Knowledge and Learning* [11].

As an author of these publications, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation, Supervision.

6.1 Summary

This chapter contributes primarily to RQ2 by focusing on designing and developing an unplugged CT activity called Cross Array Task (CAT), aimed at assessing the progression of algorithmic skills across the entire compulsory education path (K-12) in Switzerland. It begins by outlining the design of the activity, detailing its objectives and structure, followed by an explanation of how interaction strategies and algorithms can be categorised and how these classifications inform the evaluation metric used to assess AT skills. By focusing on the development of a practical and scalable assessment tool, the chapter provides insight into how the unplugged CAT can be used effectively across various educational contexts to measure AT.

6.2 The cross array

The concept of the *cross array* was developed to engage students in recognising patterns and building AT through a visually stimulating and structured activity. Visual patterns, such as the cross array, can be effective in engaging learners because they encourage the recognition of regularities, such as repetitions, symmetries, and colour patterns, which are foundational skills in AT. This approach draws on cognitive development theories, which suggest that tasks involving visual patterns are well-suited to measure domain-general components of AT, such as pattern recognition, generalisation, and the ability to articulate algorithmic procedures.

The design of the CAT was informed by the principles of situated cognition, which emphasise that learning occurs most effectively in contexts that are meaningful and contextually rich. The task complexity and progression were carefully calibrated to create a balance between accessibility for younger learners and increasing challenges to stimulate growth in AT. The task sequence begins with simpler patterns and gradually increases in complexity, allowing students to build on their existing knowledge and develop new cognitive strategies.

Initially, the design was tested with kindergarten students, as it was assumed they would provide insights into how early-stage problem-solving strategies evolve and how they engage with tasks requiring pattern recognition. Based on feedback from these initial tests, the designs were refined, resulting in 12 patterns of varying complexity, presented in Figure 6.1. Each cross-shaped design, consisting of five 2×2 square arrays of coloured dots, was selected for its simplicity and ability to introduce different patterns. The final set of 12 schemas provides a progression that supports the development of increasingly sophisticated algorithmic thinking as students advance through the tasks.

The task complexity is designed to maintain engagement and encourage problem-solving strategies that are applicable to the task at hand and transferable to other contexts. Feedback from younger learners and the complexity adjustments made during testing ensured that the tasks offered a meaningful experience, motivating students to engage deeply while refining their AT skills. With these adjustments, we hypothesise that the task will be equally effective for older students, who can engage with the same patterns using more complex strategies. By aligning task complexity with developmental stages, we aim to capture the progression in AT and assess the relationship between task difficulty and students' problem-solving strategies across different age groups.

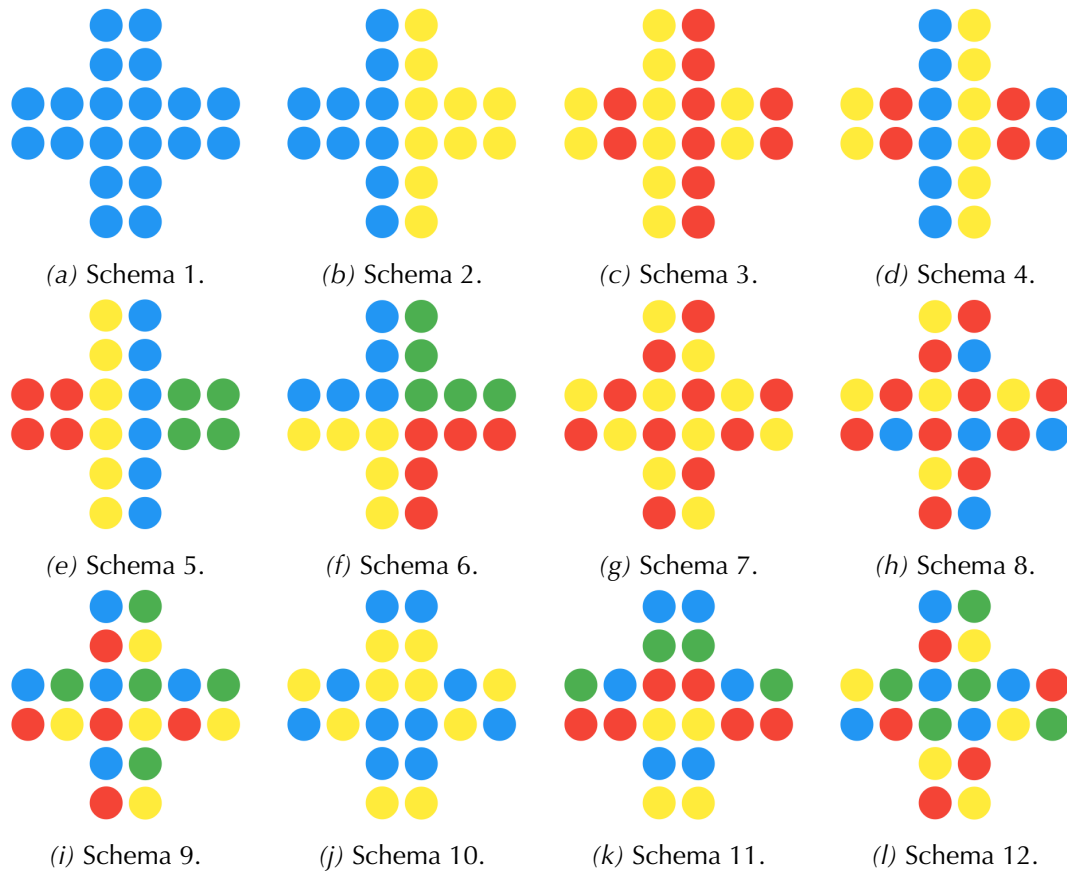


Figure 6.1 – Sequence of cross array schemas.

The figure showcases the 12 schemas proposed in the task, named from Schema 1 to Schema 12, each distinguished by its unique visual regularities and complexities, varying in elements such as colours, symmetries, alternations and other distinctive features..

6.3 Activity design

Following the methodological frameworks presented in Part II, we designed the unplugged CAT to evaluate AT.

The activity is administered in class (see Figure 6.2), and is characterised by face-to-face interaction between the pupil (problem solver), and the administrator (a human agent). The task involves the pupil observing a reference cross array and then conceptualising an algorithm to describe it to the administrator, who cannot see the original array. The goal is for the administrator to replicate the colouring pattern on a blank cross array based solely on the pupil's verbal instructions.

To begin the activity, the administrator explains the task to the pupil with the following instructions: *“You have a coloured array in front of you. I have the same array,*

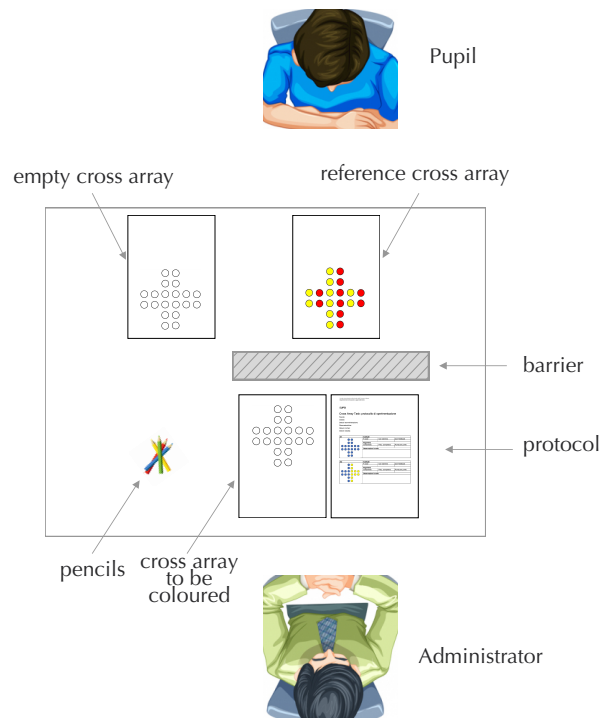


Figure 6.2 – Experimental settings (unplugged CAT).

Pupil and administrator are seated at two opposite sides of a table. The pupil is tasked with instructing the administrator in recreating a given reference cross array schema. The instructions can be communicated verbally or through gestures on a supporting empty cross array schema. Initially, the pupil cannot see the administrator's actions due to a physical barrier preventing visual cues, which can be removed upon request. The administrator interprets and records all of the pupil's instructions and algorithms in a protocol.

but uncoloured, in front of me. You should describe your array so I can colour mine the same way. You can try to describe it by voice. You can indicate the dots on the empty array on your right if it is too difficult. If it is still too difficult, you can ask me to remove the screen so that you can look at what I'm colouring."

During the activity, the administrator, in addition, to interpreting students' instructions and using them to colour the schema, records the pupil's algorithmic procedure, the artefactual environment used, and the level of autonomy exhibited during the task in a protocol for later analysis (see Appendix C.1). These elements are crucial for assessing the pupil's AT skills and are classified according to specific criteria, explained in the following sections.

6.3.1 CT-cube dimensions of the CAT

During the development of this activity, we focused on three key aspects of the CT-cube: the cognitive activity performed, the artefactual environment used, and the autonomy of the individuals involved.

Among the possible cognitive activities, the CAT targets the *algorithm* dimension, simplifying both the problem setting and the assessment components to make the task more accessible for young learners, as illustrated in Figure 6.3.

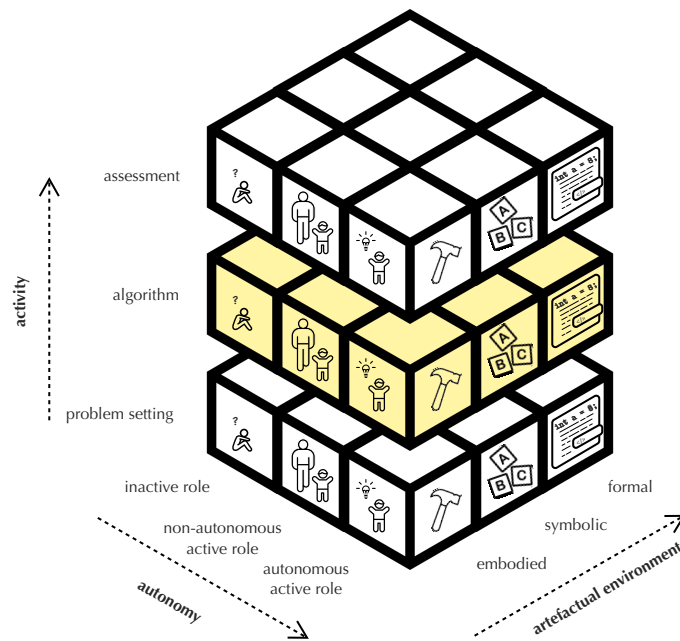


Figure 6.3 – The algorithm activity of the CT-cube.

Regarding the artefactual environment, only two dimensions are explored in the unplugged CAT, the *embodied* and *symbolic* computational worlds. The formal computational world is excluded, as it is considered too abstract for the age group targeted by the task [86, 122]. In particular, pupils can communicate their instructions verbally, describing the process using natural language. This is considered a symbolic artefact, as it relies on words and phrases to represent ideas and concepts. Alternatively, pupils can enhance their verbal instructions with physical gestures, such as pointing to specific dots on an empty cross array schema. This form of communication, which illustrates the instructions through hand movements, is considered an embodied artefact.

For the autonomy levels, all three levels of autonomy, *inactive*, *active non-autonomous*, and *active autonomous*, which reflect varying degrees of engagement and independence during the activity, are considered. Pupils are considered inactive if they

do not attempt to solve the task or are unable to provide intelligible instructions to the administrator. They are classified as non-autonomous if the barrier between them is removed and they can give intelligible instructions to the administrator. Finally, they are considered to have an autonomous active role if they provide intelligible instructions to the administrator while the barrier remains in place.

Activity states

Figure 6.4 represents the possible cells of the CT-cube for the algorithmic activity in the CAT, showing the different combinations of activity, autonomy, and artefactual environment.

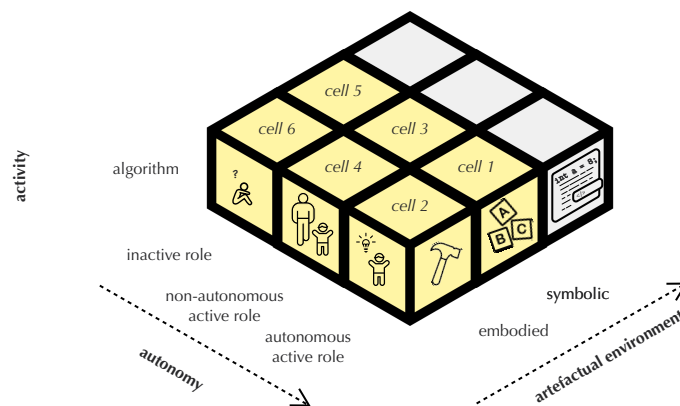


Figure 6.4 – The possible cells in the algorithm activity of the CT-cube for the CAT.

The activity starts in the CT-cube cell corresponding to *algorithm - autonomous role - symbolic artefact* (cell 1). If the pupil is inactive or is giving false or incomplete instructions trying to solve the task only by voice, corresponding to *algorithm - inactive role - symbolic artefact* (cell 5), the administrator suggests to the pupil to use the empty cross array, corresponding to *algorithm - autonomous role - embodied artefact* (cell 2). If also, in this case, the pupil is inactive or is giving false or incomplete instructions, corresponding to *algorithm - inactive role - embodied artefact* (cell 6), the administrator removes the barrier and shows to the pupil what he/she is colouring, corresponding to *algorithm - non-autonomous role - embodied artefact* (cell 4). The possibility for the pupil to observe what the administrator is doing is considered non-autonomous as it is a particular kind of (indirect) support. If also, in the latter case, the pupil is inactive or is giving false or incomplete instructions (cell 5 and cell 6), the task is finished and considered *unsuccessful*.

In each case, the pupil is free to use the empty array on his/her right or to ask to remove the screen at each moment. The task is considered *successful* if the pupil

is able to give complete and correct instructions (eventually with corrections during the description, for example, after having removed the screen) to the administrator, independently from the artefacts used by and the active role (non-autonomous or autonomous) of the pupil. In the rest of the paper, we call *algorithm* the entire set of correct instructions given by the pupil.

6.4 Activity profile

The components and characteristics of the unplugged CAT are illustrated following our FADE-CTP and graphically represented in Figure 6.5. Additionally, we provide an overview of the overall CT competencies that can be developed through this activity, as well as those that cannot, and finally, we present in Figure 6.6 the resulting profile, mapping the relationship between the activity's characteristics and the CT competencies it activates.

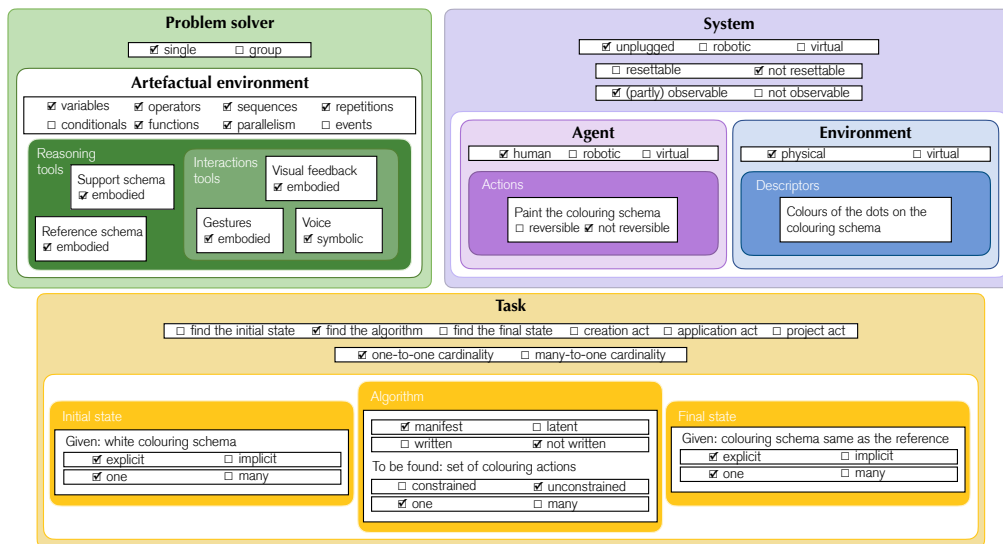


Figure 6.5 – Components and characteristics (unplugged CAT).

6.4.1 Components

- *Problem solver*: the student who has to communicate an algorithm corresponding to the sequence of instructions to reproduce the colouring of the reference schema. The artefactual environment comprises cognitive tools such as support and reference schemas, which are available to the problem solver to reason

about the task. Additionally, the problem solver can interact with the system to communicate the algorithm. This can be achieved using a natural language such as the voice (symbolic) or gestures (embodied) on the colouring schema, an empty cross array used from the agent. Moreover, by removing the screen that separates the problem solver from the agent, he can have visual feedback (embodied) of the cross array being coloured.

- *Agent*: the researcher, executor of the problem solver's instructions, responsible for filling the colouring schema according to the problem solver's algorithm. The agent's actions are not resettable.
- *Environment*: the cross array to be coloured, whose state is described by the colour of each dot (white, yellow, blue, green, or red).
- *Task*: find the algorithm. The system's state is defined by the colouring cross status, initially white and, at the end, the same as the reference schema. The algorithm is the set of agent instructions to achieve this transformation.

6.4.2 Characteristics

- *Tool functionalities*: voice and gestures provide various functionalities associated with algorithmic concepts suitable to design the algorithm, including (i) *variables* can represent different colours of the cross array dots; (ii) *operators* are used to change the colour of the dots performing actions such as colouring a dot, a row, a square and so on; (iii) *sequences* determine the order in which the actions should be executed to achieve the desired outcome; (iv) *repetitions* allow for repeating specific sequences of operations, such as colouring the first column in red and repeating it every two columns; (v) *functions* consist of operations that perform a specific task and can be applied to different inputs, for example, creating a pattern of alternating red and yellow dots in a square and applying it to different positions of the cross array; (vi) *parallelism* involves executing multiple actions simultaneously and can be associated with using symmetries to describe the pattern.
- *System resetability*: the system is not resettable since it is impossible to reverse the agent's actions.
- *System observability*: the system is partially observable since the cross array being coloured by default is not seen until the end of the task unless the problem solver demands otherwise.
- *Task cardinality*: the task has a one-to-one mapping, with given one initial and one final state, and an algorithm to be found.
- *Task explicitness*: all elements are given explicitly.
- *Task constraints*: the algorithm is unconstrained.

- *Algorithm representation*: the algorithm is represented through voice commands or gestures. It is considered manifest because it is externalised, but not written since it is not stored in a permanent format.

		Tool functionalities								System						Task							
		Variables	Operators	Sequences	Repetitions	Conditionals	Functions	Parallelism	Events	System resettable	System not resettable	System observable	System not observable	One-to-one cardinality	Many-to-one cardinality	Explicit elements	Implicit elements	Unconstrained elements	Constrained elements	Algorithm manifest	Algorithm latent	Algorithm written	Algorithm not written
		✓	✓	✓	✓	□	✓	✓	□	□	✓	✓	□	✓	□	✓	□	✓	□	✓	□	□	✓
Data collection		✓	✓						+	+	+	+	+	+	+	+	+	+	+	+	+	+	
Pattern recognition		✓	+	+	✓*	+	✓*			+	+	+	+	+	+	+	+	+	+	+	+	+	
Decomposition		✓	+	+	✓*	+	✓*	+		+			+	+	+	+	+	+	+	+	+	+	
Abstraction		✓	✓		+	+	✓				+		+	+		+	+	+	+	+	+	+	
Data representation		✓	✓		+	+	+				+		+	+		+	+	+	+	+	+	+	
Variables		✓	✓	+	+	+	+	+	+			+	+	+	+	+	+	+	+	+		+	
Operators		✓	+	✓	+	+	+	+	+			+	+	+		+	+	+	+			+	
Sequences		✓	+	+	✓	+		+				+	+			+	+	+	+			+	
Repetitions		✓	+	+	+	✓		+				+	+		+	+	+	+	+			+	
Conditionals		✗	+	+		✓			+			+	+	+	+	+	+	+	+			+	
Functions		✓	+	+	+	+	✓					+	+		+	+	+	+	+			+	
Parallelism		✓	+	+				✓						+	+	+	+	+	+			+	
Events		✗	+	+		+			✓			+			+	+	+	+	+			+	
Algorithm debugging		✗	+	+	+	+	+	+	+	✓	✗	+					+			✓	✗	✓	
System state verification		✗						+	+	✓	✗	+					+			✓	✗	✓	
Constraints validation		✗	+	+	+	+	+	+	+	✗	✗	+						✗	✓				
Optimisation		✗	+	+	+	+	+	+	+	✓	✗	+											
Generalisation		✗	✓		+	+	✓		+	✓	✗	+		+			+		+				

Figure 6.6 – Activity profile (unplugged CAT).

6.4.3 Competencies

Enabling features for competencies development

- *Problem setting*: all competencies can be activated thanks to the presence of variables, sequences, repetitions and functions in the tool functionalities. The presence of many tool functionalities, the non-resettability of the system and the algorithm representation positively affect and boost problem setting skills. The system observability supports data collection and pattern recognition. The one-to-one cardinality, in addition, stimulates decomposition. The explicit and unconstrained definition of the task elements also promotes pattern recognition, decomposition and abstraction.
- *Algorithm*: all competencies associated with the algorithmic concepts enabled by the tool functionalities, meaning variables, operators, sequences, repetitions and functions, can be activated and promote one another. The form of representation of the algorithm, the system observability, and the explicit and unconstrained

definition of the task elements further enhance these. The one-to-one cardinality helps to enhance some of these skills as well.

- *Assessment*: since the system is not resettable, no assessment skills can be developed.

Inhibiting features for competencies development

- *Conditionals and events*: non-activable as these functionalities are unavailable in the platform. A way to make conditionals available in the tool functionalities would be allowing the problem solver to change a dot colour, for example by communicating instructions such as: “if the dot is red, then colour it yellow”. By doing this, the problem solver engages with the concept of conditionals and can develop their algorithmic skills. The completion of each row in the cross array can be considered an event. The problem solver can specify that they want to fill the cross line by line, and once a line is complete, the researcher will move on to the next line. This allows the problem solver to list only the sequence of colours without repeating the instructions for where to go. The change in the environment (completing a row) triggers the researcher to move to the next row. Using conditionals and events can greatly enhance the complexity of the solutions that can be generated and help develop advanced CT skills.
- *Assessment skills*: the inability to reset the system impairs the development of the student’s skills. One possible solution to this issue is enabling the student to reset the colouring schema using a voice command. This would return the schema to its initial blank state, allowing the student to start the task from the beginning and practice their assessment skills. To develop system state verification, it is also essential to not reveal the initial or final states. Moreover, constraints should be imposed on the algorithm to develop constraint validation skills, for example, limiting the use of specific operators or the number of times they can be used, allowing the problem solver to develop the ability to think about the constraints and limitations in their algorithms.

6.5 Competencies assessment

Although the unplugged CAT activates a range of competencies, our focus is specifically on measuring algorithmic skills, while also considering aspects of situated cognition, specifically the ability to effectively use the available artefacts and the nature of the social dynamics during the task (see Figure 6.7). Additionally, we evaluated participation and success rates, providing further insights into students’ engagement and their ability to complete the task successfully.

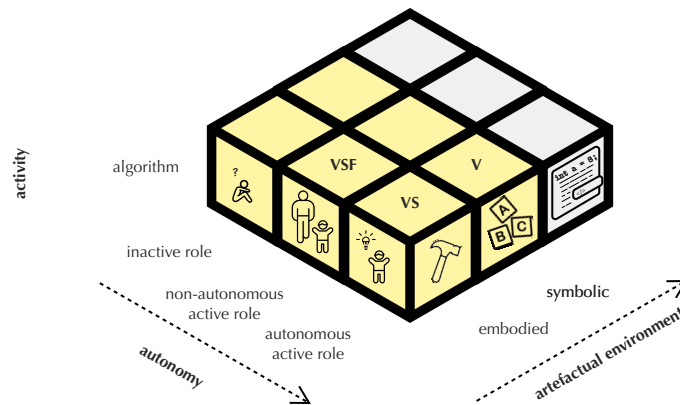


Figure 6.7 – Assessed cells in the algorithm activity of the CT-cube for the CAT.

6.5.1 Algorithm dimension

The CAT evaluates students' algorithmic skills by examining the complexity of the operations used to describe the cross arrays. Each operation has an associated level of complexity, which ranges across three distinct levels. The algorithm dimension, or classification, reflects the highest level of complexity among the operations it contains. We distinguish between:

1. *0D, zero-dimensional algorithms*: this level involves describing the cross array point by point, with each dot coloured individually using a specified colour. These operations are referred to as Colour-One-Dot operations (COD).
2. *1D, one-dimensional algorithms*: this level involves describing the cross array with a series of dots arranged in structures such as rows, columns, diagonals, squares, L-shapes, zig-zags, half-crosses, or entire crosses, all with a specified colour. These operations are referred to as Colour-Several-Dots (CSDs).
3. *2D, two-dimensional algorithms*: this level involves describing the cross array using more advanced operations, such as sequences of CSDs with alternating colours, repetitions or mirroring of COD and/or CSDs operations.

All 12 cross arrays used in our experimental study could be described using zero-, one- or two-dimensional algorithms.

An algorithm is *redundant* if one or more dots are described more than once in the algorithm. Additionally, we define the *number of operations* used in an algorithm as the total number of CODs and CSDs used in the algorithm, where a COD and/or CSD used inside a loop is considered only once. The maximal number of operations in non-redundant algorithms is 20, the minimal number is 1.

Example algorithms for describing Schema 3

Below, we present three different example algorithms for describing Schema 3 in our sequence (see Figure 6.1c).

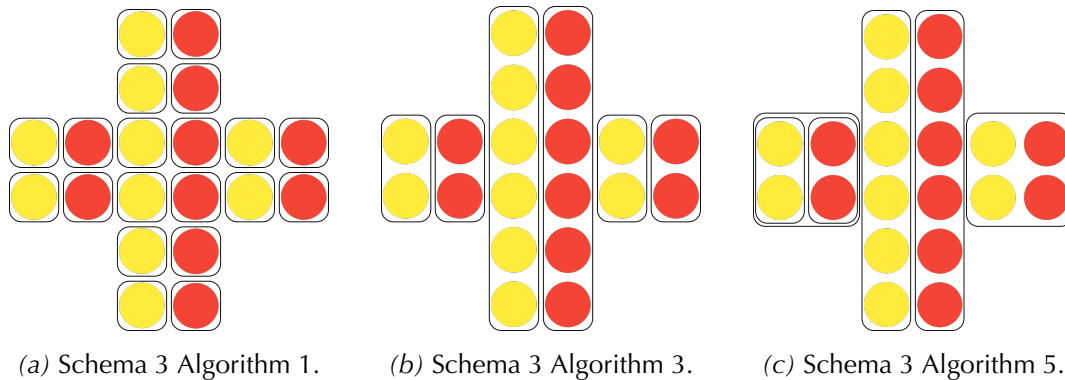


Figure 6.8 – Examples of algorithms for Schema 3.

If the pupil describes the array point by point without redundancy (see Figure 6.8a), he is using a 0D algorithm with 20 CODs (one for each dot), consequently, the number of operations is equal to 20.

If the pupil describes the array column by column without any loop (see Figure 6.8b), the algorithm consists of six CSDs (one for each column) and is 1D. The number of operations corresponds to the number of CSDs (there is no COD) and is equal to 6.

Finally, we considered an algorithm in which the left square is described through its two columns, and the right square is described as equal to the left one, while the two columns in the middle are described column by column (see Figure 6.8c). To generate

Listing 6.1 – Schema 3 Algorithm 5 pseudo-code.

```

1  # One cycle for each square (e.g., i=1 left square, i=2 right square)
2  for i from 1 to 2 do
3      # Colour the left column of the square in yellow
4      CSD(column, yellow)
5      # Colour the right column of the square in red
6      CSD(column, red)
7  end for
8  # Colour the left central column in yellow
9  CSD(column, yellow)
10 # Colour the right central column in red
11 CSD(column, red)

```

the right square, it is sufficient to repeat the operations performed to generate the left square and could be described in a pseudo-code using a loop (see Listing 6.1). In this case, the algorithm contains a loop on two CSD (see line 2), making it 2D. The number of operations is 4, corresponding to the number of CSDs, respectively, on lines 4, 6, 9 and 11 (CSDs inside a loop are considered once).

6.5.2 Interaction dimension

The CAT, in addition to algorithmic skills, evaluates aspects of situated cognition by assessing how students apply their knowledge in context, including interactions with artefacts and social dynamics. In particular, the interaction dimension reflects both the complexity of the artefacts used by the students during the task and the level of autonomy demonstrated, determined by the extent to which they asked for visual cues and relied on visual feedback. We distinguish between:

1. *VSF*: this level involves using voice and hand gestures on an empty cross array, hinging on visual feedback;
2. *VS*: this level involves using voice and hand gestures on an empty cross array, without hinging on visual feedback, and
3. *V*: this level involves using only voice without hand gestures on an empty cross array or visual feedback.

6.5.3 CAT score

The task is considered successful if the student creates a complete and correct algorithm, regardless of its complexity, the artefactual environment, or the level of autonomy. To measure how a pupil's competencies evolve, we define a single metric, the *CAT score*, which quantifies their multi-faceted performance, encompassing both the algorithm and interaction dimensions (see Table 6.1). For more details on the CAT assessment rubric refer to Section 8.2.

The algorithm dimension score ranges from 0 (for the simplest level of complexity, 0D) to 2 (for the most complex, 2D). The overall algorithm dimension score reflects the most complex operation successfully performed by the student during the assessment.

The interaction dimension score ranges from 0 (for the simplest level of complexity, VSF) to 2 (for the most complex, V). The overall interaction dimension score reflects the lowest level of effective complexity of interaction demonstrated by the student during the assessment. Specifically, the exclusive use of voice (V) is considered more complex and valuable than using voice alongside the empty array (VS). Additionally, a more autonomous role (using voice alone) is valued over a less autonomous one (relying on visual feedback).

Table 6.1 – CAT score (unplugged CAT).

Rows represent the algorithm dimensions and columns represent the interaction dimensions.

		Competence level		
		VSF	VS	V
		$c = 1$	$c = 2$	$c = 3$
Competence component	0D			
	$r = 1$	0	1	2
	1D			
	$r = 2$	1	2	3
	2D			
	$r = 3$	2	3	4

We calculate the CAT score for each task completed by the student as the sum of the two scores, ranging from 0 (minimum) to 4 (maximum). A higher score is indicative of a student who has navigated the complexities of challenging artefacts, assumed an autonomous role, and/or conceived a higher-dimensional algorithm.

The CAT score alone should not be used to compare students' performance, as different combinations of algorithm complexity and interaction strategies can result in the same total score, despite varying approaches. For example, a pupil who describes a two-dimensional algorithm using voice and an empty array (2D-VS) receives the same score as a pupil who describes a one-dimensional algorithm exclusively using voice (1D-V), even though the strategies differ in algorithm and interaction complexity. However, this does not hinder gaining a deeper understanding of students' algorithmic and interaction skills, as the separate dimensions provide detailed insights into their problem-solving approaches, even when the final score is the same.

6.5.4 Task metrics

We also considered metrics to gauge students' proficiency in AT and task execution.

Participation rate The participation rate measures whether students attempted and concluded each task assigned during the CAT assessment, regardless of correctness. Each student is assigned 12 tasks, and the participation rate indicates how many of these tasks. This metric provides an initial overview of students' engagement and persistence in the assessment activities.

Success rate The success rate evaluates the number of tasks that students correctly solved during the CAT assessment.

Chapter 7

The virtual Cross Array Task

The content of this chapter has been adapted from the following articles with permission of all co-authors and publishers:

- **Adorni, G.**, Piatti, S., and Karpenko, V. (2024). Virtual CAT: A multi-interface educational platform for algorithmic thinking assessment. *SoftwareX* [13].
- **Adorni, G.** and Piatti, A. (unpublished). Designing the virtual CAT: A digital tool for algorithmic thinking assessment in compulsory education [10].
- **Adorni, G.**, Artico, I., Piatti, A., Lutz, E., Gambardella, L. M., Negrini, L., Mondada, F., and Assaf, D. (2024). Development of algorithmic thinking skills in K-12 education: A comparative study of unplugged and digital assessment instruments. *Computers in Human Behavior Reports* [5].
- **Adorni, G.**, Piatti, A., Bumbacher, E., Negrini, L., Mondada, F., Assaf, D., Mangili, F., and Gambardella, L. M. (2025). FADE-CTP: A Framework for the Analysis and Design of Educational Computational Thinking Problems. *Technology, Knowledge and Learning* [11].

As author of these publications, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation, Supervision.

7.1 Summary

This chapter contributes primarily to RQ2, by focusing on the design and development of the virtual adaptation of the CAT, aimed at enabling automated large-scale assessment of algorithmic skills across the entire compulsory education path (K-12) in Switzerland. It begins by describing the transition from an unplugged to a virtual format, emphasising the design of the new instrument. Next, it outlines the categorisa-

tion of the new method of interaction and variations to the algorithm dimension metric, explaining their role in shaping the evaluation metric for assessing AT. Finally, a technical overview of the instrument's implementation is provided with a detailed discussion of prototype development, demonstrating how the virtual CAT can be integrated into large-scale assessment systems for use across diverse educational contexts.

7.2 Activity design

Our first experience with the unplugged CAT highlighted several limitations that made it unsuitable for large-scale assessment. Designed as a one-on-one activity, it was time-intensive and impractical for simultaneous administration. Additionally, reliance on a human administrator introduced potential inconsistencies in interpreting and delivering instructions, and it did not allow for automated assessment, further limiting its scalability and efficiency. To address these challenges, following the methodological frameworks presented in Part II, we developed a digital version of the activity, called virtual CAT [10, 13]. This adaptation streamlined the assessment process by eliminating the dependency on human administration, ensuring consistency, and allowing for large-scale automated administration.

The virtual CAT retains the core elements of the original activity: (i) the *cognitive activity* requires students to devise a set of instructions or an algorithm to replicate coloured patterns on a cross-shaped grid; (ii) the *artefactual environment* includes a variety of cognitive artefacts based on both embodiment and perception as well as symbolic representation, even though specific artefacts used may vary; (iii) the students' *autonomy* reflects their level of independence during the activity, ranging from those who do not engage with the task to those who rely on visual feedback up to those who provide clear instructions independently.

The activity is still conducted in class (see Figure 7.1), but while the unplugged CAT relied on face-to-face interaction with a human administrator, the virtual CAT relied on a virtual agent that interprets and executes the algorithms devised by students. In particular, it acts as a programming language interpreter, translating gesture interactions and visual blocks into a formal programming language that mirrors the operations used in the unplugged activity. Algorithms are automatically recorded, enabling immediate evaluation of students' strategies and providing feedback along with optional guidance or hints. This approach eliminates human errors and inconsistencies in instruction interpretation, ensuring a more standardised assessment experience.

Administration via individual devices enables simultaneous participation and assessment by multiple pupils, allowing entire classrooms or larger groups to engage with the activity independently on their devices, thereby overcoming the logistical chal-

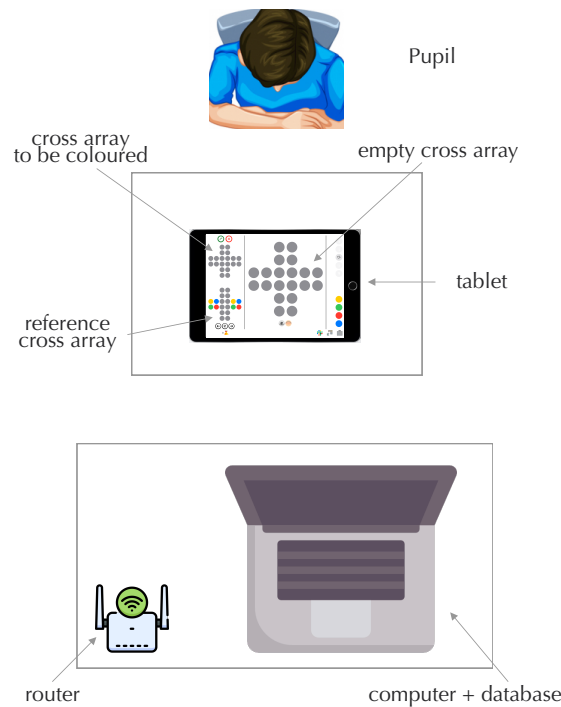


Figure 7.1 – Experimental settings (virtual CAT)

The pupil is tasked with recreating a given reference cross array schema using either a gesture-based or visual block-based programming interface. Initially, the pupil cannot view the outcome of their instructions due to a feature that blocks visual cues, which can be toggled as needed. The system automatically interprets and logs all of the student's actions and algorithms in an external database.

lenges of one-on-one interactions in the unplugged version without requiring additional human resources.

As for the artefacts available in the virtual environment, two interfaces are provided to accommodate diverse learning styles and preferences. We did not provide a modality of interaction based on verbal instruction, but we provided an alternative symbolic language, a visual block-based programming interface (CAT-VPI). This decision was driven by several technical challenges of implementing speech recognition, particularly in a multilingual classroom context with young students [17, 332]. Most speech recognition systems are trained on adult voices, making them less accurate for children, whose speech differs in pitch and tone [119, 120, 232, 345]. Additionally, there isn't enough data available to improve the accuracy of children's voices [56, 94]. While there are techniques to adjust children's voices to be more like adult voices, they don't work well enough [16, 155, 156, 274]. For these reasons, we chose a visual program-

ming interface instead, which is more reliable in this context, and its implementation requires less effort.

The CAT-VPI is designed to make coding accessible to K-12 students, including beginners with no prior programming experience. It allows students to construct colouring algorithms using drag-and-drop programming blocks, which mirror the instructions observed in the unplugged version of the activity. Nevertheless, these blocks are customisable, enabling users to adjust parameters like colour and pattern choices. This intuitive and flexible approach reduces the likelihood of syntax errors, potentially improving the overall learning experience. For a visual representation of this interface, refer to Figure D.6, with further details on its development provided in Section 7.5.3.

The CAT-GI is designed to emulate the hand gestures observed in the unplugged CAT activity, providing a tactile experience similar to interacting with the physical cross array that ensures continuity in the interaction type while leveraging digital capabilities. Users can build the colouring algorithm by selecting colours, tapping on individual dots, dragging across multiple dots to create patterns, or using icons to perform more advanced actions, such as repeating instructions or mirroring patterns. For a visual representation of this interface, refer to Figure D.8, with further details on its development provided in Section 7.5.3.

The digital version retains the challenge of limited visual feedback present in the unplugged CAT by restricting students' access to the agent's progress unless explicitly enabled, thereby preserving the original task's difficulty and autonomy while offering flexibility in feedback.

Pupils also have the flexibility to choose their preferred interaction mode, navigate between tasks, restart them, confirm completion, or skip them as needed. Upon completing all tasks, pupils are directed to a results dashboard that comprehensively summarises their performance, including visual representations of attempted tasks, scores, completion status, and time taken.

To ensure accessibility across Switzerland and broader applicability, the virtual CAT supports multiple languages, including Italian, French, and German, reflecting Switzerland's linguistic diversity, and also provides an English version to extend its potential use (see Figure D.1).

Table 7.1 summarise the principal differences between the unplugged and virtual versions of the CAT.

Table 7.1 – Differences between the unplugged and virtual CAT.

	Unplugged CAT	Virtual CAT
Interactions	Face-to-face (problem solver & human agent)	Face-to-device (problem solver & virtual agent)
Artefactual environment	Voice (symbolic) and hand gestures (embodied)	Block-based visual programming interface (symbolic) and gesture interface (embodied)
Autonomy	Removable physical barrier to enable visual feedback	Toggleable button to enable visual feedback
Algorithm classification	A human agent interprets instructions and manually codifies the algorithm	A virtual agent interprets instructions and automatically codifies the algorithm into formal programming language
Assessment	Manual	Automatic

7.3 Activity profile

As done for the unplugged CAT, the components and characteristics of the virtual CAT are illustrated following our FADE-CTP and graphically represented in Figure 7.2 Additionally, we provide an overview of the overall CT competencies that can be developed through this activity, as well as those that cannot, and finally we present in Figure 7.3 the resulting profile, mapping the relationship between the activity's characteristics and the CT competencies it activates.

7.3.1 Components

- *Problem solver*: the student who has to communicate the algorithm, or the sequence of instructions to reproduce the colouring of the reference schema. The artefactual environment comprises cognitive tools such as support and reference schemas, which are available to the problem solver to reason about the task. Additionally, the problem solver can interact with the system to communicate the algorithm. This can be achieved using a block-based visual programming interface (symbolic) or a gesture interface (embodied). Moreover, the problem solver can eventually observe the status cross array being coloured, enabling the visual feedback (embodied).
- *Agent*: the virtual agent, which automatically interprets and executes the problem solver's instructions. The agent's actions are resettable.
- *Environment*: the cross array to be coloured, whose state is described by the

colour of each dot (white, yellow, blue, green, or red).

- **Task:** find the algorithm. The system's state is defined by the colouring cross status, initially white and, at the end, the same as the reference schema. The algorithm is the set of agent instructions to achieve this transformation.

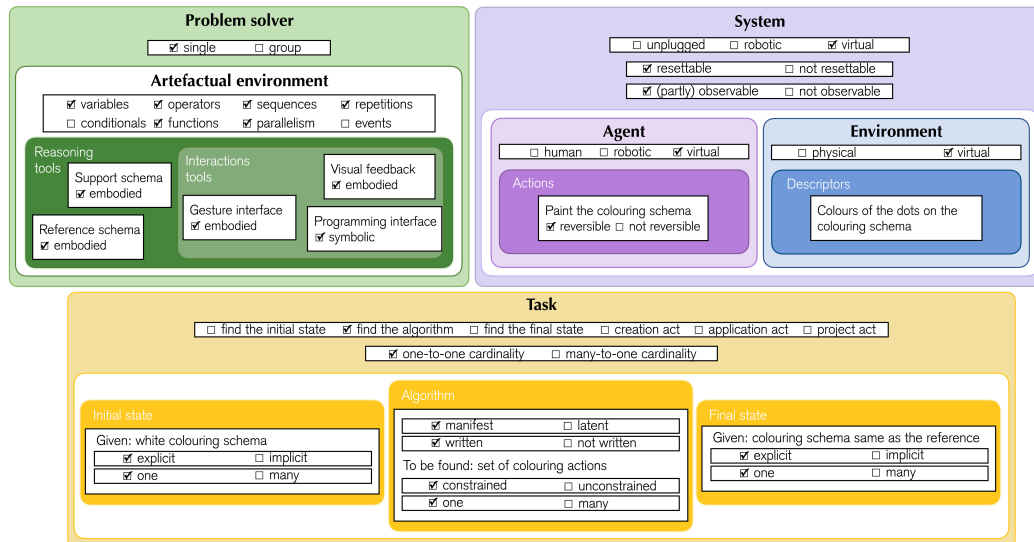


Figure 7.2 – Components and characteristics (virtual CAT).

7.3.2 Characteristics

- **Tool functionalities:** the same core functionalities associated with algorithmic concepts as the unplugged CAT are supported in the virtual CAT through the gesture and programming interfaces.
- **System resettability:** the system is instead resettable, since students can reverse the agent's actions by restarting the task or modifying the algorithm.
- **System observability:** the system remains partially observable since the cross array being coloured by default is not seen until the end of the task unless the student clicks the button to enable visual feedback.
- **Task cardinality:** the task has a one-to-one mapping, with given one initial and one final state, and an algorithm to be found.
- **Task explicitness:** all elements are given explicitly.
- **Task constraints:** the algorithm is now constrained since pupils can use only the commands made available.

- *Algorithm representation*: the algorithm is represented through visual code blocks or gestures. It remains manifest, as it is externalised, and in the case of code blocks, it is also written, since it is stored in a permanent format.

Table 7.2 summarises the key characteristics and differences between the CAT variants.

Table 7.2 – Differences between the unplugged and virtual CAT characteristics.

	Unplugged CAT	Virtual CAT
<i>Tool functionalities</i>	Variables, operators, sequences, repetitions, functions and parallelism	
<i>System resettability</i>	Not resettable	Resettable
<i>System observability</i>	Partially observable	
<i>Task cardinality</i>	One-to-one	
<i>Task explicitness</i>	Explicit	
<i>Task constraints</i>	Unconstrained	Constrained
<i>Algorithm representation</i>	Manifest non-written	Manifest written

7.3.3 Competencies

Enabling features for competencies development

- *Problem setting*: all competencies can be activated, consistently with the unplugged CAT, as the core tool functionalities required to activate these skills remain unchanged.
- *Algorithm*: the same algorithmic competencies activated in the unplugged CAT are also triggered in the virtual version, as both rely on the same tool functionalities.
- *Assessment*: unlike the unplugged version, the virtual CAT enables the activation of all assessment skills, thanks to the system's resettable nature and a manifest, written representation of the algorithm. Additionally, the inclusion of constraints activates constraint validation.

Inhibiting features for competencies development

- *Conditionals and events*: these competencies remain non-activable as these functionalities are unavailable in the platform.

		Tool functionalities							System						Task								
		Variables	Operators	Sequences	Repetitions	Conditionals	Functions	Parallelism	Events	System resettable	System not resettable	System observable	System not observable	One-to-one cardinality	Many-to-one cardinality	Explicit elements	Implicit elements	Unconstrained elements	Constrained elements	Algorithm manifest	Algorithm latent	Algorithm written	Algorithm not written
		✓	✓	✓	✓	□	✓	✓	□	✓	□	✓	□	✓	□	✓	□	□	✓	✓	□	✓	□
Data collection		✓	✓						+	+	+	+	+	+	+		+		+	+	+	+	
Pattern recognition		✓	+		+	✓*	+	✓*		+	+	+		+	+		+		+	+	+	+	
Decomposition		✓	+	+	✓*	+	+	✓*	+	+	+	+		+	+		+		+	+	+	+	
Abstraction		✓	✓		+	+	+	✓			+			+	+		+		+	+	+	+	
Data representation		✓	✓		+	+	+	+			+			+	+		+		+	+	+	+	
Variables		✓	✓	+	+	+	+	+	+			+	+	+	+	+	+	+	+	+		+	
Operators		✓	+	✓	+	+	+	+	+			+	+	+	+	+	+	+	+		+	+	
Sequences		✓	+	+	✓	+		+			+	+			+	+	+	+	+	+	+	+	
Repetitions		✓	+	+	+	✓		+			+	+			+	+	+	+	+	+	+	+	
Conditionals		✗	+	+		✓			+			+	+	+	+	+	+	+	+	+	+	+	
Functions		✓	+	+	+	+		✓			+	+			+	+	+	+			+	+	
Parallelism		✓	+	+				✓							+	+	+	+			+	+	
Events		✗	+	+		+			✓			+				+		+	+	+	+	+	
Algorithm debugging		✓	+	+	+	+	+	+	+	✓	✗	+					+			✓	✗	✓	
System state verification		✓								✓	✗	+				+			✓	✗	✓	✗	
Constraints validation		✓	+	+	+	+	+	+	+	✓	✗	+					✗	✓					
Optimisation		✓	+	+	+	+	+	+	+	✓	✗	+											
Generalisation		✓	✓		+	+	+	✓	+	✓	✗	+		+		+		+					

Figure 7.3 – Activity profile (virtual CAT).

7.4 Competencies assessment

As in the unplugged version, the primary focus of the virtual CAT is on measuring algorithmic skills while considering aspects of situated cognition, participation rate, and success rate (see Figure 7.4). However, the digital environment introduces an addi-

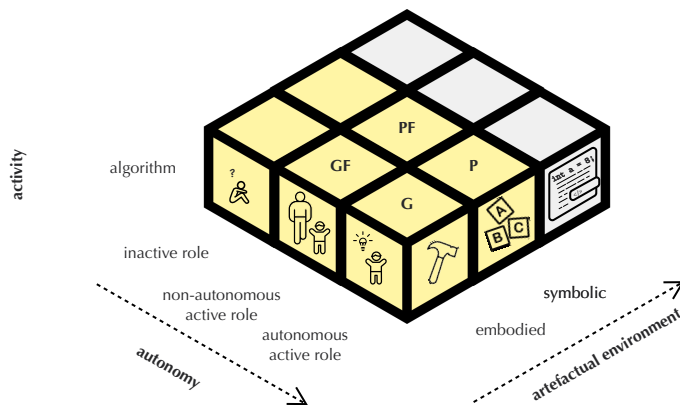


Figure 7.4 – Assessed cells in the algorithm activity of the CT-cube for the virtual CAT.

tional layer of assessment, such as evaluating students' proficiency in task execution.

This includes quantifying trial-and-error strategies through the number of restarts and assessing efficiency based on the time taken to solve the tasks.

7.4.1 Algorithm dimension

The algorithm dimensions in the virtual CAT remain consistent with those in the unplugged version, evaluating students' algorithmic skills based on the complexity of the operations used to describe the cross arrays (*0D*, *1D*, and *2D* algorithms).

7.4.2 Adjusted algorithm dimension

Acknowledging the need to assess algorithm efficiency alongside complexity, we have introduced an adapted metric that considers the number of commands used, providing a more nuanced evaluation of students' algorithmic competencies. It recognises cases where a simpler yet more efficient algorithm may perform better than a complex one with more commands. The adjusted score, denoted as \widetilde{AD} , is calculated using a formula that balances the highest complexity level achieved by the student against the overall workload:

$$\widetilde{AD} = \frac{1 + P_{\max-d} + \sum_d (C_d \cdot P_d)}{C_{\text{total}}}, \quad (7.1)$$

where, d is the complexity level of the algorithm (i.e., 0, 1, or 2); $P_{\max-d}$ are the points assigned to the highest complexity level used by the student, computed as the original algorithm dimension score plus one; C_d is the number of commands at complexity level d ; P_d are the points for the complexity level d , computed as the original algorithm dimension score at that complexity level plus one; C_{total} is the overall number of commands used across all levels. The first term in Equation (7.1) gives a score for the most complex algorithm achieved by the student, adjusted for the total commands used, favouring higher-level algorithms but considering the overall workload in terms of the number of commands executed. The second term calculates a weighted score for each complexity level, factoring in the proportion of commands used at each complexity level relative to the total command count and multiplying it by the points for that level.

7.4.3 Interaction dimension

The interaction dimension in the Virtual CAT expands to include four levels of complexity, reflecting both the type of interface used and the level of reliance on visual feedback. We distinguish between:

- GF*: this level involves using the gesture interface, hinging on visual feedback;
G: this level involves using the gesture interface, without hinging on visual feedback;
PF: this level involves using the visual programming interface, hinging on visual feedback;
P: this level involves using the visual programming interface, without hinging on visual feedback;

7.4.4 CAT score

The CAT score for the virtual CAT (see Table 7.3) is computed following the same approach as the unplugged version, with a key difference in the interaction dimension score, which now ranges from 0 (for the simplest level of complexity, GF) to 3 (for the most complex, P). The overall CAT score for each task completed by the student is calculated as the sum of the two dimension scores, ranging from 0 (minimum) to 5 (maximum). For more detail on the development of unplugged CAT assessment rubric refer to Section 8.2.

Table 7.3 – CAT score (virtual CAT).

Rows represent the algorithm dimensions and columns represent the interaction dimensions.

		Competence level			
		GF	G	PF	P
		<i>c</i> = 1	<i>c</i> = 2	<i>c</i> = 3	<i>c</i> = 4
Competence component	0D <i>r</i> = 1	0	1	2	3
	1D <i>r</i> = 2	1	2	3	4
	2D <i>r</i> = 3	2	3	4	5

7.4.5 Task metrics

Additionally, the CAT assessment instrument evaluates various metrics to gauge students' proficiency in AT and task execution.

Participation rate The participation rate measures whether students attempted and concluded each task assigned during the CAT assessment, regardless of correctness. Each student is assigned 12 tasks, and the participation rate indicates how many of

these tasks. This metric provides an initial overview of students' engagement and persistence in the assessment activities.

Success rate The success rate evaluates the number of tasks that students correctly solved during the CAT assessment, irrespective of efficiency or the number of attempts made.

Number of restarts The number of restarts reflects students' approach to problem-solving, particularly their use of trial and error (T&E) strategies. It counts instances where students choose to restart the tasks, indicating their iterative approach to refining algorithms and achieving desired outcomes.

Efficiency Efficiency evaluates how effectively students complete tasks, considering the time taken as a factor.

7.5 Instrument development and implementation

In this section, we present our instrument development strategy, guided by the User Experience (UX) design life cycle. This structured approach, involving the systematic collection of data on user behaviours, preferences, and requirements, ensures the development of a user-centred product aligned with their actual needs [123]. Additionally, we discuss the technical components and architectural choices behind the platform's development, covering its framework, data management, programming language formalization, and interpreter implementation.

7.5.1 Development process

The UX design life cycle is an iterative process that encompasses three main phases: (1) *understand* (U) – gathering insights into user needs and problem domains; (2) *make* (M) – designing and prototyping solutions based on the understanding phase; (3) *evaluate* (E) – testing prototypes and solutions through user feedback and expert analysis [123]. The cyclic flow between understanding, making, and evaluating emphasises the iterative nature of this process. Insights gained from evaluations often lead to revisiting earlier phases to refine and enhance the design. While the specific phases and iterations of the UX design life cycle can vary in the literature, we have adopted the process illustrated in Figure 7.5 to ensure a structured and user-centred approach.

In the context of product development, two types of evaluations are commonly employed to guide and assess design: formative and summative evaluation. *Formative*

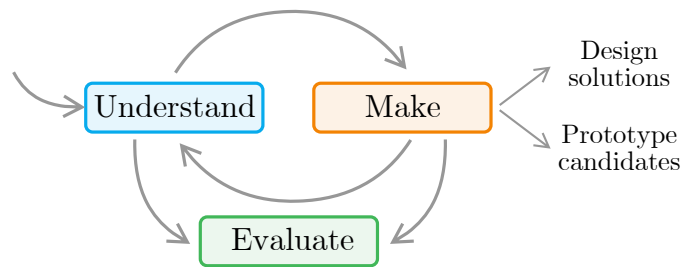


Figure 7.5 – UX design life cycle.

This process encompasses understanding user needs (U), making (M) – or designing and prototyping solutions –, and evaluating them through user and expert feedback (E). These phases repeat cyclically, with evaluation insights leading to refinements in earlier stages.

evaluation takes place throughout the iterative design process, helping to refine and improve the product before it reaches its final form. It ensures that continuous user feedback informs ongoing refinements of the design, intending to improve usability, functionality, and overall user experience. *Summative evaluation*, on the other hand, is conducted once the product has been fully developed, aiming to assess its overall effectiveness and impact. It occurs after the design has been finalised. It focuses primarily on evaluating the effectiveness and impact of the final product, typically through large-scale studies, and does not involve redesign or further iterations [48, 82, 267, 331].

Our design and evaluation process, illustrated in Figure 7.6, followed a structured and iterative procedure aligned with the phases of the UX design cycle of Figure 7.5. The process spanned 17 months, from February 2022 to June 2023. The first prototype was developed from February to July 2022 (5 months), the second from July 2022 to March 2023 (8 months), and the final version from March to June 2023 (4 months).

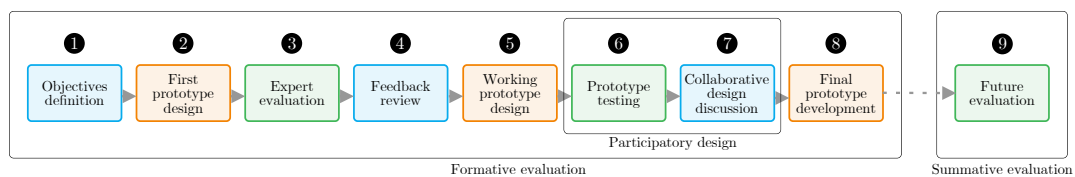


Figure 7.6 – Design and evaluation process overview.

The various stages of the design process, from defining objectives to developing and evaluating prototypes. Different colours represent the phases of the UX design lifecycle: blue for the understand (U) phase, orange for the make (M) phase, and green for the evaluate (E) phase.

Objectives definition

This process began with the objectives definition stage ❶, where we set the instrument's goals and decided how to adapt the unplugged activity to the virtual format. As discussed in Section 7.2, we focused on preserving the pedagogical value of the original task by identifying key components to maintain and determining how to translate them into the digital environment. This step was crucial for ensuring the educational objectives were upheld, laying the foundation for the entire process.

Initial prototype development

The second step in the process focuses on developing the initial digital prototype ❷, prioritising user experience accessibility and usability [123]. In educational technology, accessibility centres on crafting solutions to meet users' needs from various backgrounds, regardless of their physical or cognitive abilities [153]. In contrast, usability focuses on the user experience, aiming at delivering an intuitive and effective learning environment [15, 105]. To achieve these objectives, we made several key decisions, including selecting the devices on which the tool would be available, determining supported languages, and defining the layout of the user interfaces, all in accordance with established guidelines and best practices [123].

The development of the initial prototype was grounded in the architectural and technical decisions outlined in Section 7.5.2, ensuring the prototype's functionality and alignment with the educational objectives defined in Section 7.5.1. In this phase, we sketched interface layouts, selected appropriate technologies, and built interactive prototypes to simulate user interactions. Throughout this process, we continuously evaluated the user experience, ensuring the prototype was intuitive for students and teachers. Additionally, expert consultations were integrated to validate the design choices, ensuring the prototype met usability and pedagogical standards.

Due to time constraints and limited access to schools and children, we skipped certain prototyping stages, such as producing paper prototypes, and directly developed a functional prototype. This streamlined approach was also necessary because of the age of the children involved in the participatory design, who may struggle with abstract reasoning and therefore require a more accessible prototype [86, 118, 122, 183]. The resulting first prototype is detailed in Section 7.5.3.

Expert evaluation and prototype redesign

Following, we conducted an expert evaluation ❸ to assess the prototype design, usability and accessibility [123]. This step involved the participation of experts in both UX design and educational technology, who examined the prototype and provided detailed

feedback on various aspects of the platform, particularly focusing on interface clarity, functionality, and alignment with educational objectives. The first expert consulted, recruited through our institutional network, is an interaction design teacher-researcher with a background in educational technology and user-centred design. The expert was provided with a brief description of the platform's intended use before independently exploring the application. His feedback was collected during a collaborative session, in which he shared detailed observations after testing the platform. In addition, three pedagogical experts with experience in computer science education were invited to review the prototype. These professionals were selected based on recommendations and their known contributions to technology-enhanced learning. After being introduced to the platform, each expert independently tested the application and subsequently shared their observations during a feedback session.

During the reflective phase ④, we carefully analysed and prioritised the changes proposed by the experts, ensuring that the adjustments aligned with both usability principles and pedagogical goals.

The prototype redesign ⑤ incorporated these changes and included the development of key technical features, such as a virtual interpreter and the infrastructure necessary for real-time interaction and data processing. The feedback received, the modifications decided upon, and the resulting updated prototype are documented in Section 7.5.3.

Participatory design

Following the expert evaluations and the creation of the second prototype, the next phase of the design process focused on participatory design and the development of the final application. This phase began with engaging the target users, students and teachers, in testing the prototype in real-world settings and providing feedback on its usability and effectiveness ⑥. The goal was to integrate their insights and preferences to refine the platform, ensuring it met their educational needs and user requirements. Our pilot study was designed as a participatory process involving three key roles: a researcher from our team, students and teachers [265]. Details about the selection and participation of students and teachers, as well as the experimental setting and administration procedures, are provided in Section 10.2.2.

User feedback elicitation During the validation phase, we collected feedback from both students and teachers to evaluate their experience with the tool and identify usability issues, thus refining the tool to meet user needs.

Pupils were at the heart of the study, and their interactions with the platform were crucial for assessing the tool's usability and identifying new user requirements

[87, 168, 265, 310, 311]. We actively engaged children as informants and evaluators, enabling us to design with their needs and preferences in mind [109, 110, 239, 240]. Their evolving thoughts and reflections, shared during testing activities, provided real-time insights into how they perceived and interacted with the tool [135, 142, 143, 343]. This participatory approach empowered children to take ownership of the tool's development while fostering critical thinking about its features [141, 144, 158, 159]. It also ensured the process remained enjoyable and rewarding for them, aligning with principles of co-design and participatory research [39, 200, 201, 202].

Teachers played an essential role by facilitating the study and providing assistance as needed [24, 40, 86, 108, 131, 149, 206, 207, 263]. These teachers were the ones present in the classrooms during the activities and were responsible for ensuring a smooth classroom experience. Their observations highlighted how students engaged with the tool, identified areas of difficulty, and noted moments of success. Teachers' feedback was invaluable in refining the platform to balance educational goals with practical usability and address both pedagogical and logistical challenges in the classroom.

During the study, the administrator from our research team closely monitored pupil progress and interactions collected empirical data on task performance and gathered feedback from students and teachers. Multiple data elicitation techniques, including think-aloud and observation, have been employed to gain insights into the usability and effectiveness of the design [121, 123]. In particular, students' interactions with the tool have been documented, focusing on their behaviours, verbal feedback, and non-verbal cues. Key observations included moments of confusion, problem-solving strategies, and how students navigated specific features [118]. Real-time note-taking captured recurring patterns and usability issues, providing valuable insights into user experience. This structured approach ensured a detailed understanding of the tool's strengths and areas for improvement, directly informing subsequent design iterations [109, 121, 123]. These techniques, grounded in Human-Computer Interaction (HCI) and UX design principles, enabled us to triangulate data sources and derive actionable insights for iterative improvements [88, 121, 123, 135, 173, 182].

Collaborative design At the end of this process, we conducted a collaborative session with the users ⑦, during which design proposals were presented based on the notes and feedback collected by the researcher. Users were also invited to provide additional input. Based on their insights, modifications were proposed and discussed, enabling users to actively contribute to refining the prototype and ensuring the design better aligns with their needs. In Section 7.5.3, we highlight the feedback received during this collaborative session.

Final application development

In the final phase, the prototype is redesigned ⑧ in response to the feedback and suggestions from the collaborative session, leading to the final working version, which is documented in Section 7.5.3. The redesign adhered to standard mobile application design principles to enhance usability and accessibility [62, 123, 204, 298].

To create an interface familiar to the user, we incorporated common elements, like a top bar and a left-side menu list. Legibility and readability were prioritised using large font sizes and ensuring a high contrast between text and background. Accessibility considerations were central to the redesign. The interface included a colour-blind mode, high-contrast visuals, and a text-to-speech feature to accommodate users with visual impairments.

Consistency was maintained by using uniform names and labels for similar objects and functions, avoiding synonyms to ensure clarity and reduce cognitive load. Frequently used features were placed in easily accessible locations, aligning with common mobile application conventions. By adhering to these principles, the final application aimed to provide a user-friendly and inclusive experience for a diverse range of users.

The results concerning the formative evaluation are discussed in Chapter 9. The final summative evaluation ⑨, aimed at assessing the effectiveness and impact of the final working prototype on a larger scale, is discussed in Chapter 11.

7.5.2 Implementation

In terms of platform selection, we focused on iPads as the primary target device. This choice was driven by the device's user-friendly touchscreen interface, which aligns with our goal of creating an intuitive, interactive learning experience, particularly for students in K-12 educational settings [263]. Additionally, the portability and wide adoption of iPads in educational contexts made them an ideal choice for our application, ensuring the instrument would be accessible to a broad range of students.

The application was developed using the Flutter framework, selected for its robust capabilities and cross-platform support [103]. This framework enables the creation of a single codebase that operates seamlessly across multiple platforms, including Android, iOS, Linux, macOS, Windows, and web. This approach significantly streamlined development efforts and reduced the time required for platform-specific customisation.

Although the application was primarily designed for iPads, its responsive design ensures a consistent and engaging user experience across various devices and screen sizes. This flexibility guarantees that the application functions effectively without compromising on design or usability.

A key feature of Flutter that enhanced our development workflow was its hot reload

functionality, which allowed real-time previews of code changes. This feature proved essential in supporting the iterative design process, increasing both efficiency and productivity. Furthermore, Flutter’s rich library of pre-built widgets and tools made it easier to develop visually appealing and interactive user interfaces.

The latest version of the virtual CAT application, including its full source code and comprehensive documentation, is openly accessible online [12]. For a detailed explanation of the system’s data infrastructure, including collection and transfer protocols, application features, development process, and intended use cases, refer to the dedicated software paper [13].

CAT programming language

To establish a standardised set of instructions that users could employ within the application interfaces to design the algorithm, we defined the CAT programming language, which codifies and formalises all the commands and actions observed during the original experimental study with the unplugged CAT. The detailed list of commands available in the formal CAT programming language is provided in Table 7.4.

Cross representation The cross-board dots are manipulated and referenced using a coordinate system (see Figure 7.7), where rows are labelled from bottom to top using letters (A-F), and columns are numbered from left to right (1-6).

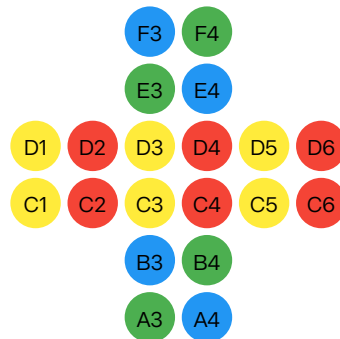


Figure 7.7 – Example of a cross-board with coordinate labels.

Moves Moving around the cross-board can be done in two ways (see Figure 7.8): the `goCell(cell)` method allows jumping directly to a specific coordinate; the `go(move, repetitions)` method allows traversing a certain number of dots in one of the eight available directions (cardinal or diagonal) to reach the desired destination.

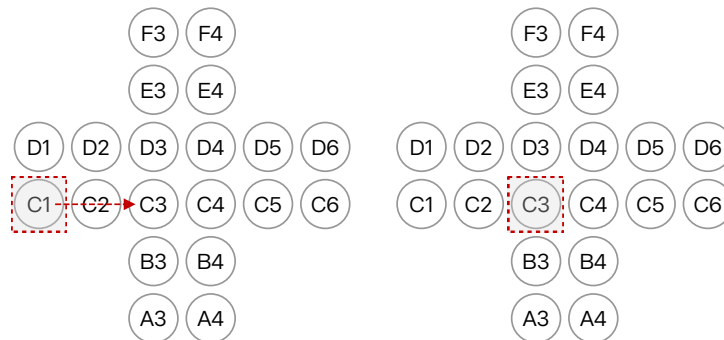


Figure 7.8 – Example of movement on the cross-board.

Starting from C1, the destination cell C3 can be reached either by using the `goCell(C3)` command or by traversing two steps to the right using the `go(right,2)` command.

Basic colouring Colouring the board can be achieved through various methods (see Figure 7.9). The `paintSingleCell(color)` method allows colouring the dot they are currently positioned on with a single colour. The `paintPattern(colors, repetitions, pattern)` method allows colouring multiple dots according to pre-defined patterns. A sequence of colours can be specified, which will alternate following the selected pattern. Additionally, users can choose from five pattern types (cardinal, diagonal, square, L, zigzag), each with various directions. The `paintMultipleCells(colors, cellsPositions)` method enables colouring multiple dots with custom patterns, defined by specifying the coordinates of the cells to be coloured. The `fillEmpty(color)` method colours all the uncoloured dots on the board with the same colour.

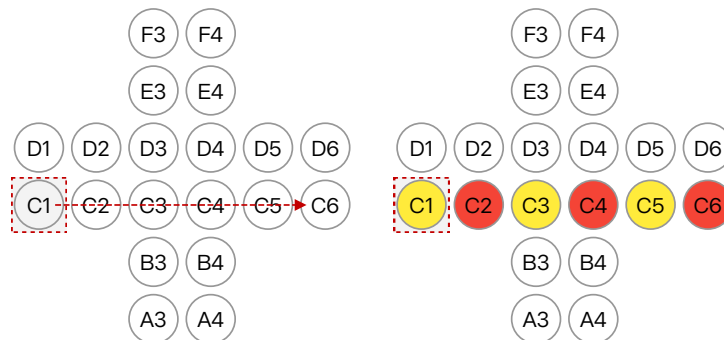


Figure 7.9 – Example of colouring a row of six dots.

Starting from C1, the row is coloured alternating yellow and red using either the `paintPattern({yellow, red}, 6, right)` or the `paintMultipleCells({yellow, red}, {C1, C2, C3, C4, C5, C6})` command.

Table 7.4 – CAT programming language commands index.

Category	Command	Description
Movements	<code>goCell(String cell)</code>	Move directly to a specific cell coordinate on the board.
	<code>go(String move, int repetitions)</code>	Moves in one of the eight possible directions, including cardinal and diagonal movements, of a specified number of cells.
Colouring	<code>paintSingleCell(String color)</code>	Colour the current cells with the specified colour.
	<code>paintPattern(List<String> colors, String repetitions, String pattern)</code>	Colour multiple cells, starting from the current, according to predefined patterns (cardinal, diagonal, square, L, zigzag) with various directions.*
	<code>paintMultipleCells(List<String> colors, List<String> cellsPositions)</code>	Colour multiple dots, at specific coordinates, with custom patterns.*
	<code>fillEmpty(String color)</code>	Colour all uncoloured dots on the board uniformly with the specified colour.
Loops	<code>repeatCommands(List<String> commands, List<String> positions)</code>	Repeats a sequence of commands (e.g., a series of go and paint) at specified coordinates.
	<code>copyCells(List<String> origin, List<String> destination)</code>	Copies colours from one set of coordinates (origin) to another (destination).
Symmetry	<code>mirrorBoard(String direction)</code>	Reflects the coloured dots on the board onto the non-coloured ones, in accordance with the principle of symmetry, along the specified direction (horizontally on the x-axis or vertically on the y-axis).
	<code>mirrorCells(List<String> cells, String direction)</code>	Applies symmetry to specific dots across a specified direction.
	<code>mirrorCommands(List<String> commands, String direction)</code>	Reflects a sequence of commands across a specified direction.

* If a sequence of colours is specified, they will alternate following the selected pattern.

Repetition-based colouring Moving beyond the basics, other methods allow for more complex operations, like repetitions (see Figure 7.10). The `repeatCommands(commands, positions)` method allows specifying a sequence of commands (e.g., a series of go and paint operations) and applying them to specific coordinates. The `copyCells(origin, destination)` method copies the colours from origin coordinates to destination coordinates.

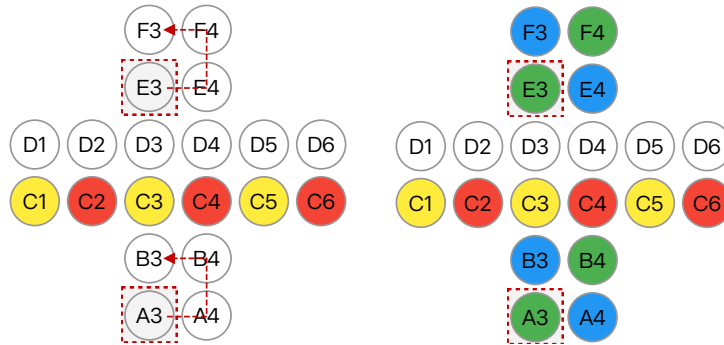


Figure 7.10 – Example of repetition of a square pattern.

On cells A3 and E3 is coloured a square pattern with alternating green and blue dots using the `repeatCommands({paintPattern({green, blue}, 4, square_right_up_left)}, {A3, E3})` command.

Symmetry-based colouring Finally, symmetrical colouring approaches are available (see Figure 7.11). The `mirrorBoard(direction)` method, which reflects the

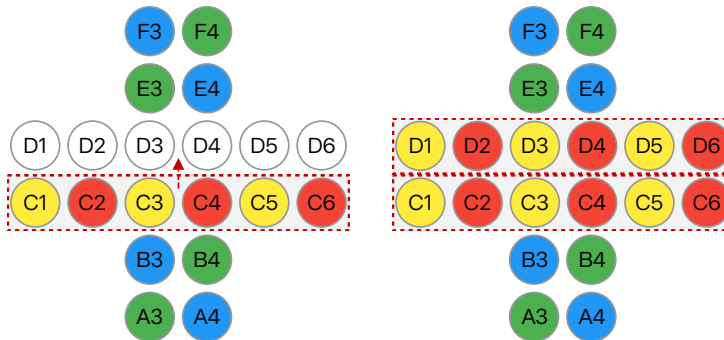


Figure 7.11 – Example of cells mirroring.

Starting from C1, all dots of the row are mirrored upwards along the horizontal axis using the `mirrorCells({C1, C2, C3, C4, C5, C6}, horizontal)` command.

coloured dots on the board onto the non-coloured ones, follows the principle of symmetry. This mirroring can be done horizontally on the x-axis or vertically on the y-axis.

The `mirrorCells(cells, direction)` method performs similar mirroring operations but on a specified set of dots. The `mirrorCommands(commands, direction)` method applies the mirroring to a list of commands.

CAT programming language interpreter

The virtual CAT programming language interpreter [8] is a dedicated Dart package that can be integrated into any Flutter project, in our case, the virtual CAT app [12]. It translates student actions, including gesture interactions and arranged visual programming blocks, into executable machine-readable instructions. It analyses the user's input, converting actions into a formal algorithm specified using the CAT programming language.

Each command that composes the algorithm, such as colour selections and other operations, undergoes a validation process to identify and address semantic errors. Notably, the interface's design, featuring predefined programming blocks and buttons, obviates the need for syntax checking, as it inherently eliminates the possibility of such errors, significantly streamlining the process. However, semantic errors can still occur during command execution, for instance, when users attempt to move outside the board boundaries using invalid directions or apply an inappropriate pattern for a colouring command.

Upon validation, the code is executed, and real-time feedback is provided to the user, including the display of current progress on the colouring cross and the CAT score. If the interpreter detects errors, it handles them and provides users with error notifications and potential suggestions for correction.

Data infrastructure

Considering the often limited availability of secure networks in educational settings, in line with the strict privacy and security demands of Swiss educational environments, we developed a secure data infrastructure that ensures a safe flow of data during the collection and assessment process between the devices used and the central data collection hub [7].

To achieve this, we established a dedicated local network by deploying a router that interconnects all the participant devices, and directs the information to a central computer that acts as the nerve centre for data collection, ensuring that the data remains within a controlled and safeguarded environment.

Gradle is an integral part of this framework, acting as the ignition system for our data process. It initiates the server and establishes the necessary network connections, enabling a seamless data exchange from the user devices to our central database. This

setup allows the data to be synchronised in real-time, ensuring that all information is current and accurately reflected across the system.

To further enhance reliability, in our system's final version, we incorporated offline functionality and automatic progress saving. These features provide flexibility by allowing students to continue their work even during technical issues, such as server disconnections, and offer more flexible time management options during assessments. This ensures that students' progress is preserved and can be resumed without data loss, mitigating disruptions to the learning and testing process.

We leverage the H2 database system for data management and organisation. In particular, Java is used for its database engine execution, and SQL is used for data manipulation and interaction.

The database schema, illustrated in Figure 7.12, features interconnected tables designed for specific roles within our educational assessment framework. Notably, the ALGORITHMS table catalogues algorithms based on complexity levels and specific commands, contributing to the algorithm dimension metric. Simultaneously, to compute the interaction dimension metric, the RESULTS table records information such as the artefact type and the level of autonomy.

Our database comprises several interconnected tables, each serving a specific purpose within our educational assessment framework. The CANTONS table lists all 26 Swiss cantons, each uniquely identified by an integer and a name with abbreviations. The SCHOOLS table details the institutions participating in the activity, linking each school with its canton. SESSIONS captures each educational session's unique characteristics, including supervisor details, school affiliation, grade level, date, and language of instruction. STUDENTS maintains essential demographic (age and gender) and session-related information for individual students. The ALGORITHMS table catalogues various algorithms by detailing their complexity levels and the presence of particular algorithmic commands, such as painting in certain patterns or directions and using colours in different ways. RESULTS table holds the performance data from user tasks, recording information like the type of interaction interface used, whether the task was completed, and evaluates the accuracy of the tasks by noting the number and correctness of dots coloured. Lastly, the LOGS table tracks user activities during tasks in great detail. The data, stored as a JSON object, includes a time-stamped record of users' actions, the commands they use, the interface they interact with, and whether visual feedback was enabled. This information allows us to track the total time users spend on tasks.

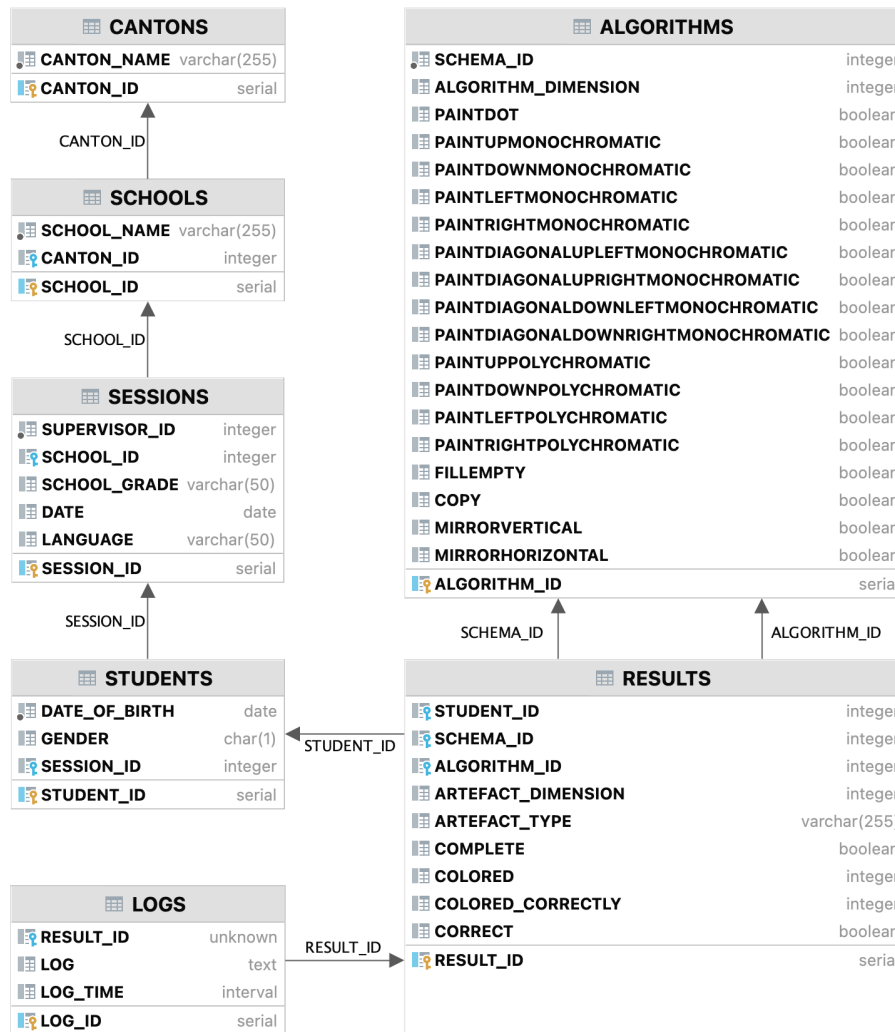


Figure 7.12 – Database schema.

The diagram depicts the logical structure of the database, including tables such as CANTONS, SCHOOLS, SESSIONS, STUDENTS, ALGORITHMS, RESULTS, and LOGS, along with their attributes and the relationships between these entities.

7.5.3 Prototypes

In this section, we present the three prototypes developed throughout the study: the initial prototype, the version refined after expert evaluation, and the final version of the application following the participatory study.

First prototype

The first prototype of the application was developed to explore and test the system's core functionalities, with the goal of creating a foundational version that experts could evaluate to gather feedback for improving its design and usability.

The CAT-VPI, illustrated in Figure 7.13, features a three-column layout. The left column **1** includes predefined code blocks, divided into two types: containers, in purple, are the commands defined in Section 7.5.2 (i.e., go to, paint, fill empty, copy, and mirror); and components, in orange, that are the inputs for container blocks, such as the colour to be used, the cell to move to or to colour, or the direction for movement or colouring. The central column **2** is the main workspace where users interact with and assemble code blocks. The right column **3** displays the reference schema to be replicated on top and the colouring schema on the bottom. This section also includes an eye icon to activate visual feedback and a green arrow to proceed to the next schema.

The CAT-GI, illustrated in Figure 7.14, presents a different layout. The bottom left section **1** contains buttons for interaction, including four selectable colours and the key commands defined in Section 7.5.2 (i.e., fill empty, copy, and two types of mirror). The right section **2** is the main workspace, featuring a large cross array that students interact with after selecting colours and/or commands. To the right of the cross is a green tick to confirm the completion of a colouring action, an eye icon above it to activate visual feedback, and a green arrow at the bottom to proceed to the next task. Finally, the top left section **3** displays the reference schema to be replicated.

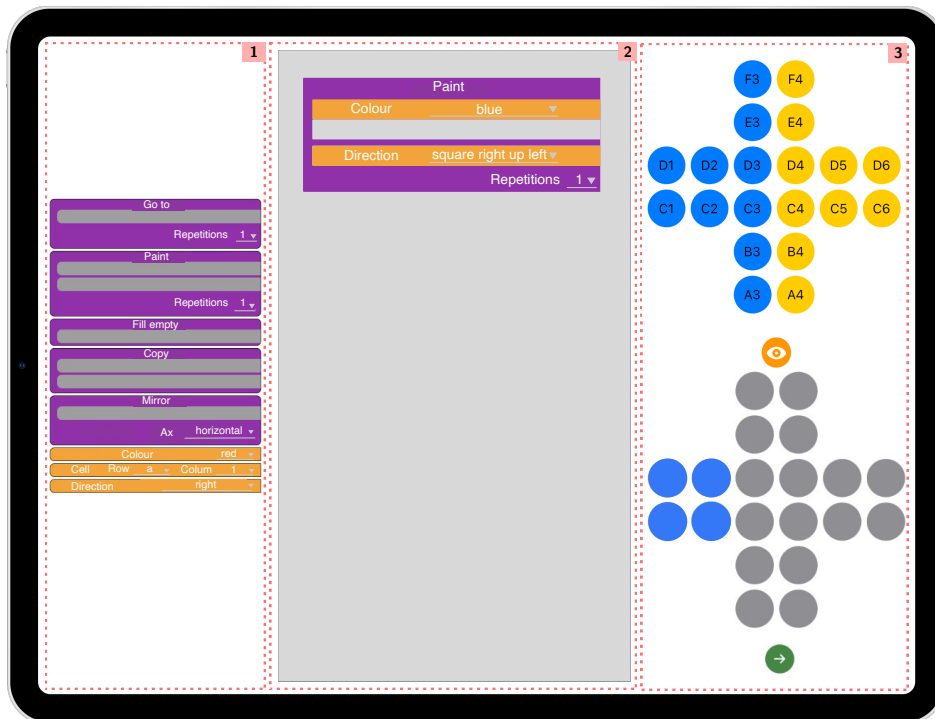


Figure 7.13 – First prototype of the CAT-VPI.

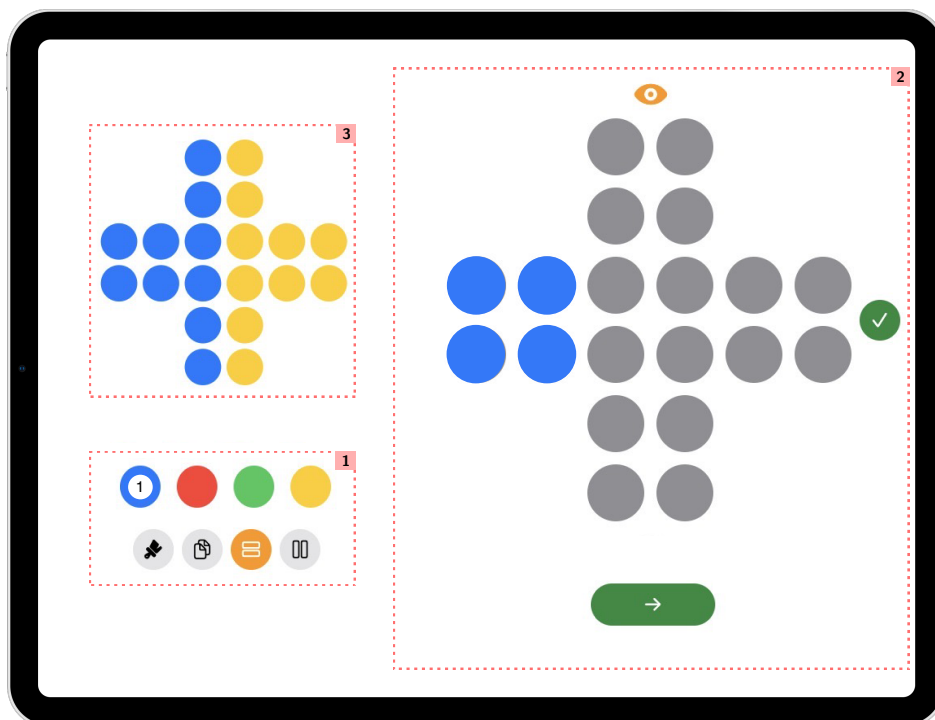


Figure 7.14 – First prototype of the CAT-GI.

Second prototype

Following expert evaluation and detailed feedback from UX and pedagogical experts, a series of modifications were made to improve the initial prototype.

UX expert feedback The feedback from the UX expert provided valuable insights into both the strengths and areas for improvement in the prototype. On the positive side, the expert highlighted the overall clarity of the platform's purpose and its potential to engage users with minimal prior experience. He also appreciated the visual layout, particularly the effectiveness of the workspace design in fostering user engagement.

However, the expert identified two key areas for improvement. First, the interface lacked consistency across different screens, which could confuse users. He recommended restructuring the interface to create a more uniform and cohesive design across interaction modalities. Second, he suggested adopting more intuitive icons to enhance the visual clarity and usability of the interface, particularly for users with limited prior experience in using such tools.

Pedagogical experts feedback The feedback from the pedagogical experts provided valuable insights into how the platform aligns with educational goals, highlighting areas for improvement to enhance its pedagogical effectiveness. The experts praised the platform's visual engagement and the thoughtful integration of AT concepts, acknowledging the design's clarity and its potential for supporting learning. However, they also identified several areas for refinement to further align the platform with best practices in computer science education.

The experts emphasised the educational benefits of allowing students to experiment, make mistakes, and learn from failures. They believe this iterative "trial and error" process fosters deeper learning and understanding. To support this, they recommended incorporating a mechanism that encourages "trial and error" preventing students from becoming discouraged by early failures. While "trial and error" can be a valuable strategy, the experts raised concerns about students potentially getting stuck in a loop without making meaningful progress. Thus, they also recommended complementing this mechanism with a system to detect when students repeatedly fail or remain inactive for long periods. This would provide targeted hints or guidance to help students reflect on their approach and adjust their strategies, ensuring they continue to move forward without becoming discouraged. Additionally, one of the recommendations was to ensure the platform is suitable for all ages. In particular, they suggested revisiting the CAT-VPI interface, as it could be too complex for younger students due to the programming blocks and the amount of text to read.

Prototype revision Figures 7.15 and 7.16 illustrates the interface layout of the second prototype. To transition to a working prototype for classroom testing, all interfaces now include three buttons at the top centre of the workspace **2** that allow users to switch between interaction modes.

In response to the feedback from the UX expert, we redesigned the user interfaces to ensure consistency between them by adopting the same three-column layout used in the CAT-VPI, illustrated in Figure 7.15. The changes were applied solely to the CAT-GI, as shown in Figure 7.16. The predefined buttons to select colours and actions are now grouped in the left column **1**. Previously, the large cross served as both the workspace and the colouring schema. Now, the central section **2** functions as the main workspace, while the right column **3** displays the reference schema at the top and the colouring schema at the bottom, where users can enable visual feedback. Additionally, new action buttons were added to the CAT-GI, aligning it with the commands available in the CAT-VPI. For example, the “copy/repeat” command, absent in the first prototype, has been included, ensuring both interfaces now offer the same set of functionalities.

A major overhaul was conducted to replace the existing icons with more intuitive and universally recognisable symbols. In the CAT-VPI, the command blocks were simplified for greater clarity. The predefined building blocks now use a colour-coding system that groups similar commands together (e.g., indigo for colouring action, orange for the mirror function, etc.). Most container blocks now come pre-loaded with the necessary components inside, so students don’t need to decide which components to include. For example, in the case of the paint block, students no longer need to figure out whether to insert the colour or another component, as these are already provided, and they only need to select the component’s specific detail, such as the colour. Instructions are provided to guide the student when a component is not pre-loaded. Other mechanisms were simplified to make the tool more intuitive, streamlining the approach by reducing the steps required for the task and improving the user experience. For example, in the previous prototype, for colouring patterns with alternating colours, users had to insert multiple colour components into the paint container and specify the number of repetitions, or cells, to colour. Now, a dedicated block is available for this operation, where users can select the colours, specify the number of repetitions, and choose a pattern.

In response to the recommendation on accessibility for younger students, we introduced two types of blocks in the CAT-VPI: textual and symbolic. While only the textual version is shown here, both types are included in Appendix D.1 for the final version of the application. Since most kindergarten pupils cannot yet read, this dual approach accommodates a wider range of users. Textual blocks provide instructions for those who can read, while symbolic blocks use intuitive symbols, offering a language-independent way to interact with the system. This ensures the platform remains accessible to younger, multilingual, or pre-literate students, making the interface engaging

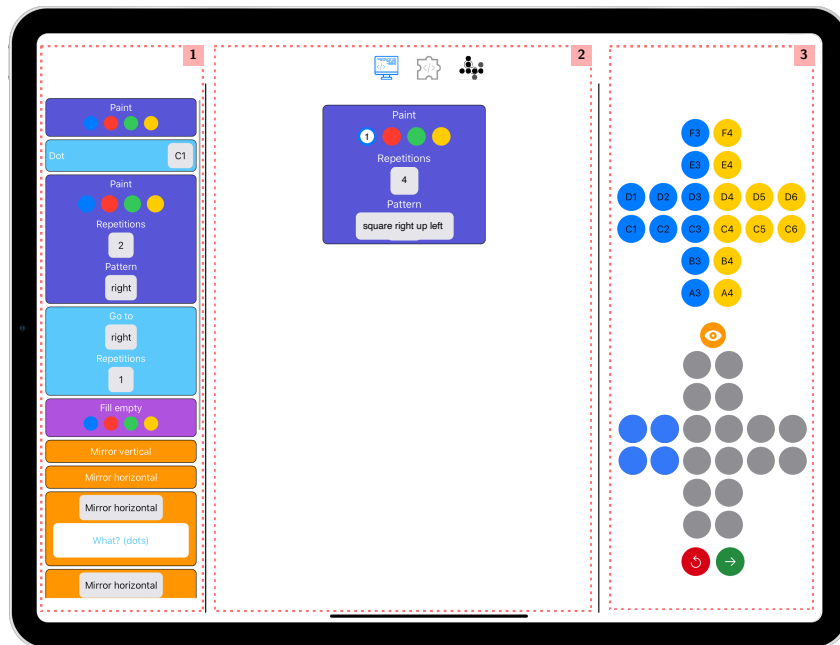


Figure 7.15 – Second prototype of the CAT-VPI.

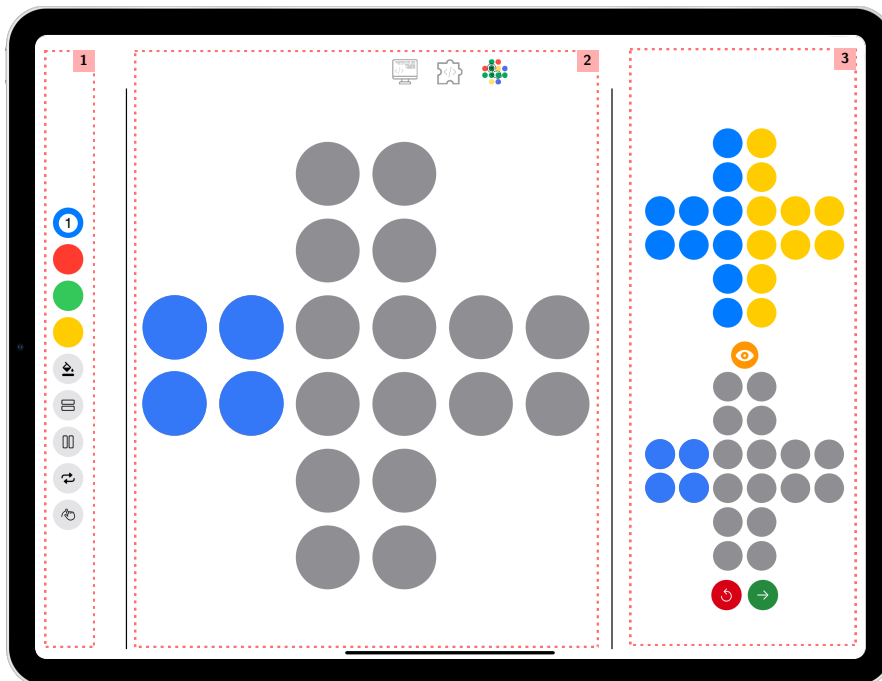


Figure 7.16 – Second prototype of the CAT-GI.

for all learners.

Finally, based on feedback from pedagogical experts regarding the “trial and error” process, we implemented a “retry” button, represented by a red circular arrow at the bottom of the colouring schema in the right section of the interfaces **3**. This feature allows students to restart exercises anytime, encouraging them to revisit their mistakes, refine their solutions, and engage in iterative learning.

Final application

The third and final version of the application was developed through active collaboration with teachers and pupils during the pilot session of the participatory study. Figures 7.17 and 7.18 illustrate the final versions of the two interfaces. Screenshots of all application screens are available in Appendix D.1, providing a comprehensive visual reference for the platform’s design and functionality.

After active collaboration with teachers and pupils from different age groups and schools during the pilot session of the participatory study. Feedback and observations of user interactions guided targeted refinements to the interface and functionality, addressing usability issues, enhancing accessibility, and better aligning the platform with the needs of students and educators.

Collaborating with students yielded invaluable insights that guided several critical changes to the platform. Initially, we observed that as time was running out, the need to confirm each schema at the end of the activity individually became cumbersome and unnecessary. Moreover, this process led to schemas being incorrectly marked as “failed” instead of “not attempted”. To address this, we introduced a “surrender” button at the bottom of the right section of the interfaces **3**, allowing users to skip specific schemas. This feature also proved useful for students who felt stuck and wanted to move on.

Another modification stemmed from feedback about the visual feedback button, which some students found unclear. In response, we replaced the original button with two new icons: an open eye indicating active visual feedback and a closed eye symbolising that feedback was turned off.

As students progressed through the activity, some expressed interest in knowing how many schemas remained to be completed. To address this, we added a progress bar at the top of the central column in the interface **4**. This addition serves a dual purpose: it satisfies students’ curiosity about their progress and resolves a limitation observed in the unplugged CAT activity, where the rigid sequencing of tasks restricted students’ ability to navigate exercises flexibly. Additionally, we included navigation arrows at the top of the right section of the interface **3** to allow students to explore upcoming schemas, providing further flexibility and accommodating those who wish to skip ahead.

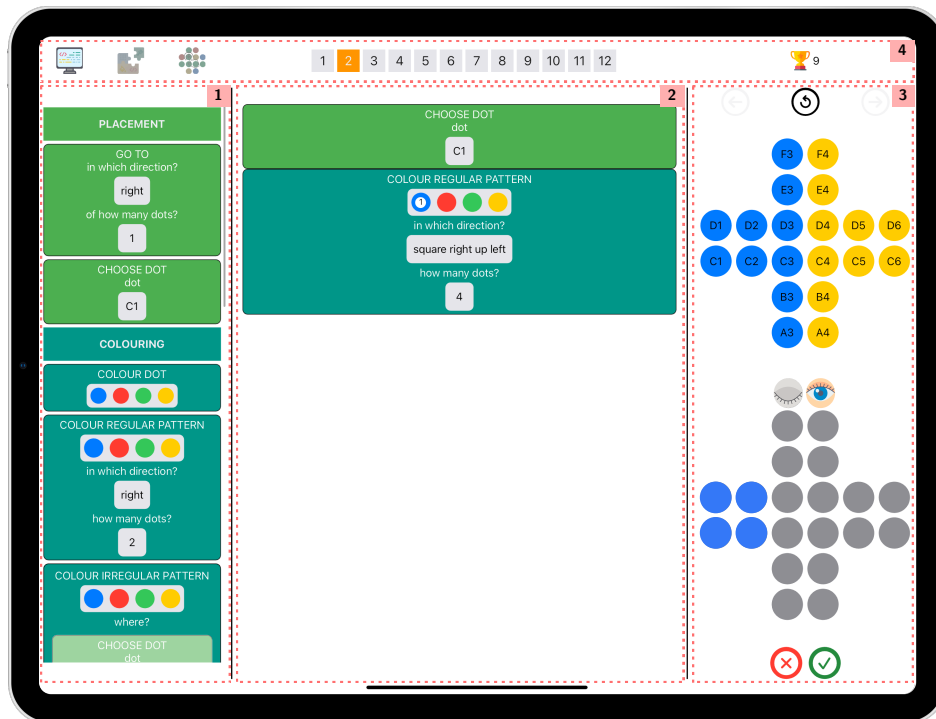


Figure 7.17 – Final CAT-VPI.

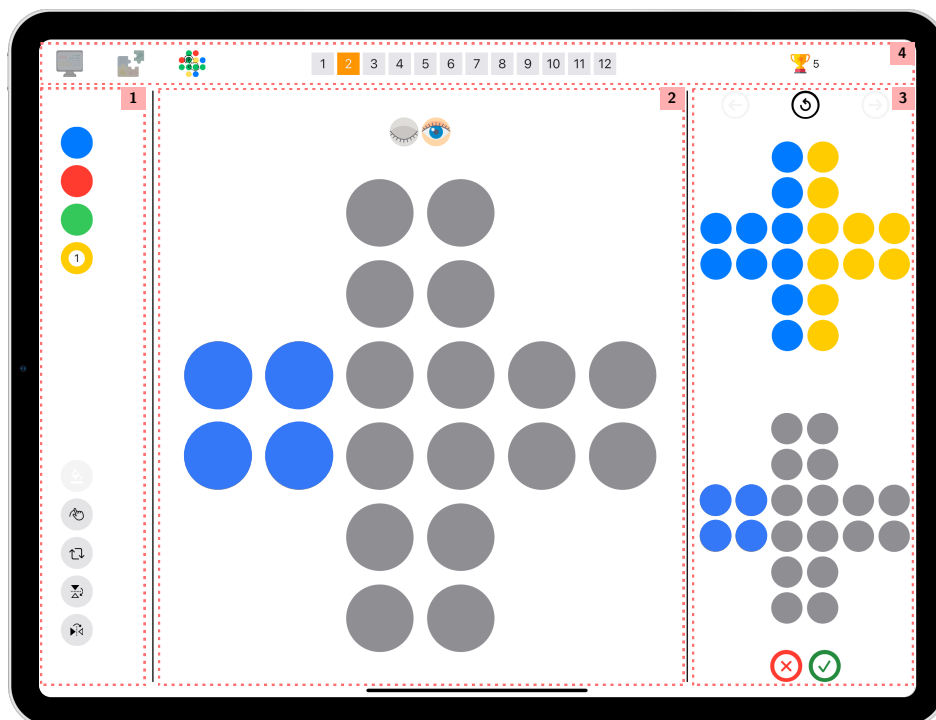


Figure 7.18 – Final CAT-GI.

Further adjustments were made based on the researcher's observations during the study, which highlighted areas for improvement that were not always evident through feedback alone. While observing pupils interacting with the CAT-VPI, it became evident that they were not using all the available commands but only readily visible ones. This was because some commands were not immediately accessible and required scrolling down the column to see them. To address this, we grouped related commands into menus in the left column of the interface [1](#) and revised their colours to improve visibility and accessibility.

Additionally, we observed that some pupils occasionally forgot to select the colour parameter within the paint blocks. Thus, we enclosed all customisable parameters within shaded boxes to make it easier for users to identify and adjust them.

Another observation concerned the use of nested blocks. Despite written instructions, some users struggled to fill these blocks correctly. To improve clarity, we added a transparent representation of the block types that could be inserted within nested blocks and provided more detailed instructions for each label.

Finally, while observing users interact with the CAT-GI, we noticed issues with certain commands, such as the `fillEmpty` button, which was often used without selecting a colour first. To address this, we implemented conditional activation of buttons, enabling them only when appropriate for the given context. Additionally, we introduced a visual feedback mechanism, including a shaking effect on incorrect actions and flashing available commands when users deviate from the intended workflow. This feature aims to guide users towards the correct actions, improving the overall user experience.

Following teacher feedback, additional improvements were made to enhance the platform further, focusing on refining the user experience and ensuring the tool met pedagogical and functional needs. One key suggestion from the teachers was to provide real-time feedback, allowing students to monitor their progress and performance during the activity. In response, in the right part of the top bar [4](#), we included a display box showing the current score for the ongoing schema.

Additionally, we introduced a final dashboard that provides a comprehensive summary of student performance across all completed schemas (see Figure D.9). This feature not only aids teachers in quickly assessing student progress but also provides students with a clear overview of their performance, helping them identify areas where they may need to focus more effort.

The initial structure of the training module required a human administrator to guide students through the platform before starting the actual validation session. This poses challenges for large-scale implementation, as reliance on this approach could introduce inconsistencies in the explanations given to different student groups, potentially affecting performance. To address this limitation and enhance the platform's scalability for broader use, we redesigned the training module by integrating standard-

ised in-app video tutorials, enabling users to navigate the platform independently (see Figure D.3). This ensures consistent instructions for all users, minimises potential biases introduced by varying researcher-led explanations, and supports more efficient large-scale data collection and assessment.

Our vision for future developments involves continuous refinement and expansion of the platform. To assess user experience, we decided to incorporate a brief survey at the end of the validation module to gather pupils' subjective impressions and insights into their perceptions of the tool (see Figure D.10). This survey aligns with established UX design techniques for data elicitation [121, 123] and the Technology Acceptance Model [123, 283], assessing factors such as ease of use, perceived usefulness, attitude towards use, and behavioural intention to assess users' acceptance of a system. It explores various facets of user interaction, from the clarity of app rules and preferred interaction modes to the perceived difficulty of exercises and overall enjoyment. Additionally, it prompts participants to reflect on whether they would use the app again in the future. To accommodate the diverse literacy levels and age groups of our users, the survey features an audio playback option for reading questions aloud, ensuring accessibility even for younger students. Responses are collected using a smiley meter scale (happy, neutral, sad), a child-friendly format shown to be effective in assessing children's attitudes toward interactive technologies [102, 118, 239].

Chapter 8

The Intelligent Assessment System for the Cross Array Task

The content of this chapter has been adapted from the following articles with permission of all co-authors and publishers:

- Antonucci, A., Mangili, F., Bonesana, C., and **Adorni, G.** (2022). Intelligent Tutoring Systems by Bayesian Nets with Noisy Gates. *The International FLAIRS Conference Proceedings* [20].
- Mangili, F., **Adorni, G.**, Piatti, A., Bonesana, C., and Antonucci, A. (2022). Modelling Assessment Rubrics through Bayesian Networks: a Pragmatic Approach. In *2022 International Conference on Software, Telecommunications and Computer Networks* (SoftCOM) [181].
- **Adorni, G.**, Mangili, F., Piatti, A., Bonesana, C., and Antonucci, A. (2023a). Rubric-based Learner Modelling via Noisy Gates Bayesian Networks for Computational Thinking Skills Assessment. *Journal of Communications Software and Systems* [9].

As an author of these publications, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation.

8.1 Summary

This chapter focuses on the definition of the IAS specific to the CAT, contributing to RQ3. It outlines the modelling of the assessment rubrics for both versions of the CAT, discussing the ordering of competencies, the encoding of answers, and the inclusion of supplementary competencies. Additionally, the chapter covers the parameter elicitation process, explaining how the necessary parameters for the model are determined,

ensuring the IAS aligns with the intended assessment of AT across different educational contexts. This process highlights how a probabilistic IAS can be effectively integrated into the CAT framework to assess AT skills in a scalable and contextually relevant way.

8.2 Modelling the CAT assessment rubric

As specified in Section 5.4, we defined a task-specific assessment rubric, for both variants of the CAT, unplugged and virtual (see Table 8.1). The instruction sequences conceived by the pupils, called algorithms, are ranked into three categories corresponding to the assessment rubric's competence components (rows). Each row represents the pupil's ability to solve a CAT schema using a certain algorithmic dimension. The interaction dimension of the pupils, given by their degree of autonomy and the tools used to accomplish the task, have been hierarchically ordered from the highest (right) to lowest (left) and determine the competence levels in the columns of the rubric.

Table 8.1 – Definition of the CAT assessment rubric.

(a) Unplugged CAT.				(b) Virtual CAT.						
		Competence level					Competence level			
		VSF	VS	V			GF	G	PF	P
		$c = 1$	$c = 2$	$c = 3$			$c = 1$	$c = 2$	$c = 3$	$c = 4$
Competence component	0D				Competence component	0D				
	$r = 1$	X_{11}	X_{12}	X_{13}		$r = 1$	X_{11}	X_{12}	X_{13}	X_{14}
	1D					1D				
	$r = 2$	X_{21}	X_{22}	X_{23}		$r = 2$	X_{21}	X_{22}	X_{23}	X_{24}
	2D					2D				
	$r = 3$	X_{31}	X_{32}	X_{33}		$r = 3$	X_{31}	X_{32}	X_{33}	X_{34}

Based on our task-specific assessment rubric, we developed a learner model, as described in Section 5.2. Figure 8.1 illustrates a simplification unplugged CAT model represented as a BN with noisy gates. The model includes 9 latent skill nodes (X_{rc}) representing the competencies from the rubric, 108 answer nodes (Y_{rc}^t) corresponding to manifest behaviours (9 skills \times 12 tasks), and 9 inhibitor nodes (X'_{rc}) representing the skill states. The network uses noisy gates to model the probability that a skill does not contribute to a specific behaviour or task outcome, providing a probabilistic framework to capture the influence of skills on performance.

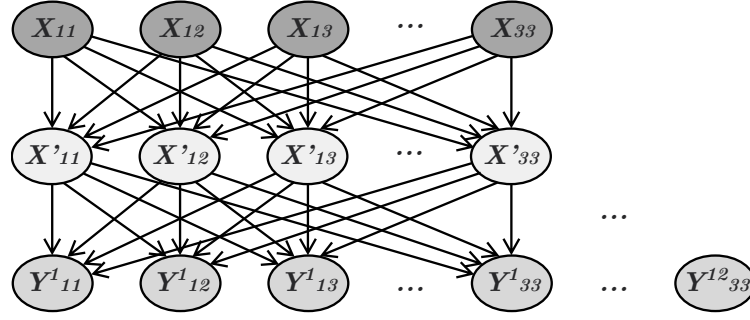


Figure 8.1 – Example of a noisy gates BN modelling the unplugged CAT assessment rubric.

8.2.1 Ordering of competencies

As introduced in Section 5.5, the columns of an assessment rubric provide the competence levels in increasing order from left to right. Sometimes, as in this case study, this is true also for the rows, where competence components are ordered from the lower (0D at the top) to the highest (2D at the bottom). This follows from the assumption that mastering algorithms of higher complexity implies also mastering simpler ones. The same is valid for the interaction dimension.

Summing up, we can conclude that a competence level X_{rc} is higher than $X_{r'c'}$ whenever $c > c'$ and $r \geq r'$, or $c = c'$ and $r > r'$. When, instead, $c > c'$ but $r < r'$, neither skill can be said to dominate the other. From the CAT assessment rubric in Table 8.1, we define a set of n target skills to be assessed: 9 for the unplugged CAT and 12 for the virtual CAT.

Accordingly, with the method described in Section 5.5, a latent skill node X_{rc} is included in the BN learner model for each of the n target skills of the rubric. The hierarchy of competencies is then modelled by n latent binary variables $D_{rc,r'c'}$, as described in Section 5.4, encoding the implication $X_{rc} \implies X_{r'c'}$ for each pair of consecutive skills in the hierarchy, i.e., such that $(r = r' + 1) \wedge (c = c')$ or $(r = r') \wedge (c = c' + 1)$.

Also, the BN includes an observable answer node Y_{rc}^t for each skill in the rubric and each task $t = 1, \dots, 12$ in the sequence of 12 similar tasks administered during the CAT experiments. Observing $Y_{rc}^t = 1$ means that the pupil has solved the t -th CAT schema using an algorithm of complexity corresponding to the c -th row of the rubric and requesting help in the r -th column. By way of example, in the unplugged CAT, a student solving the t -th schema conceiving a 0D algorithm using voice, empty schema and feedback (0D-VSF) results in the observed node $Y_{11}^t = 1$.

Figure 8.2 illustrates the noisy gates BN for the unplugged CAT case study updated

to account for competencies order by including constraints.

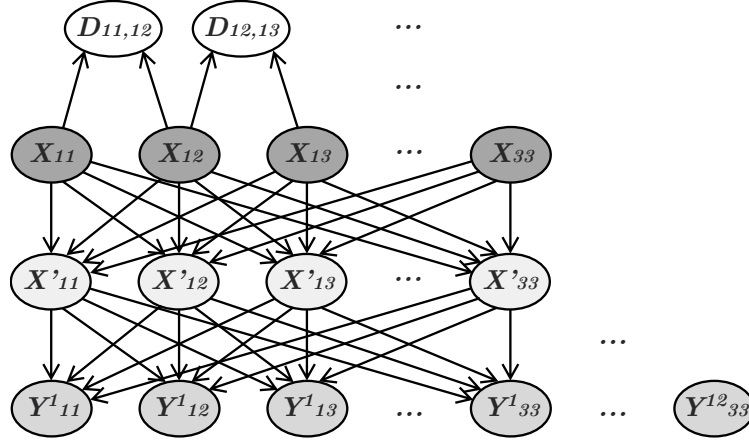


Figure 8.2 – Example of a constrained noisy gates BN modelling the unplugged CAT assessment rubric.

8.2.2 Answers encoding

In principle, all answer nodes should be explicitly observed through specific interactions with the pupil. However, this is impossible for the specific activity, as the pupils are free to choose their preferred solving approach during the CAT. Therefore, to make the answers of the pupils in this activity compatible with our model, we encoded them as follows: a task t solved at level c^* by an algorithm with complexity r^* was translated into $Y_{rc}^t = 1$ for all competence levels rc lower than or equal to r^*c^* , thus assuming that, if requested, the pupil would have been able to implement solutions requiring a lower competence level than the one used. Similarly, we set all answer nodes $Y_{rc}^t = 0$ for all higher levels, leaving those not directly comparable unobserved.

As an example, Table 8.2a illustrates the case in which the pupil engaged in the unplugged CAT has generated as a solution for task t a one-dimensional algorithm using only the empty schema and the voice (1D-VS). This choice also contributed to stressing the ordering of skills.

However, since in the extended model, the ordering of variables is modelled by explicit constraints imposed through the auxiliary variables $D_{rc,r'c'}$, such a choice would be unnecessary and detrimental, as it would artificially multiply the number of observations. Therefore, in the constrained model, a task t solved at level c^* by an algorithm with complexity r^* would be better translated into the single observation $Y_{r^*c^*}^t = 1$.

Since in the experimental setting of the CAT activity, pupils were always allowed to try solving the task with the lowest competence level (0D-VSF for the unplugged task

Table 8.2 – Example of answer encoding for 1D-VS in the unplugged CAT.

Assuming a pupil has generated a 1D-VS solution for the t -th schema: $Y_{22}^t = 1$, the two tables illustrated the answer encodings for the unconstrained and constrained cases. The symbol \emptyset indicates that the answer node is not observed.

(a) Unconstrained learner model.		Competence level		
Competence component		VSF	VS	V
		$c = 1$	$c = 2$	$c = 3$
0D	$r = 1$	1	1	\emptyset
1D	$r = 2$	1	1	0
2D	$r = 3$	\emptyset	0	0

(b) Constrained learner model.		Competence level		
Competence component		VSF	VS	V
		$c = 1$	$c = 2$	$c = 3$
0D	$r = 1$	\emptyset	\emptyset	\emptyset
1D	$r = 2$	\emptyset	1	0
2D	$r = 3$	\emptyset	0	\emptyset

and 0D-GF for the virtual task), a failure could only be observed for that level, with the consequence that only answer nodes Y_{11}^t can be directly observed in the false state $Y_{11}^t = 0$. To work around this problem, we set the answer nodes just above the one observed in the true state, i.e., $Y_{r^*(c^*+1)}^t$ and $Y_{(r^*+1)c^*}^t$ to the false state, leaving all other nodes unobserved. Table 8.2b shows how the answer encoding changes in the case of a 1D-VS solution to task t for the constrained model of the unplugged CAT.

8.2.3 Supplementary competencies

Finally, we observed that additional skills beyond those defined in the assessment rubric may be necessary depending on the specific CAT schema, especially for 1D and 2D algorithms. Through an analysis of the structures and characteristics of the CAT schemas, we identified three groups of supplementary skills: 10 for the unplugged CAT and 14 for the virtual CAT. These supplementary skills were added as new nodes to the skill network. In particular, the first group contains only one skill, represented by the variable S_1 , which is essential to implement 0D algorithms, the paint single dot operation. The second group comprises skills required for monochromatic structures, which are associated with 1D algorithms, represented by variables S_2 to S_7 for the unplugged CAT and S_2 to S_8 for the virtual CAT. The third group includes skills necessary for handling polychromatic structures and for operations such as repeating and mirroring a structure associated with 2D algorithms, represented by variables S_8 to S_{10} for the unplugged CAT and S_9 to S_{14} for the virtual CAT.

From the data collected during the experimental study with both the unplugged and virtual CAT, it was possible to extract direct observations about using each supple-

mentary skill in each task. Consequently, answer nodes $Y_{S_i}^t$ were added to the network for each task $t = 1, \dots, 12$ and each supplementary skill S_i , with $i = 1, \dots, 10$ from the unplugged CAT and $i = 1, \dots, 14$ for the virtual CAT. Each schema can be solved using one or more supplementary skills, but using all of them is not always possible. Answer nodes $Y_{S_i}^t$ take the value one if the pupil has used the i -th supplementary skill in the solution of CAT schema t , and zero otherwise.

As described in Section 5.5, a noisy-OR combines the variables in the same group into the group auxiliary nodes G_i , with $i = 1, \dots, 4$, where G_4 combines the target skills X_{rc} . In contrast, the relation between the group nodes and the target skills is conveyed through the logical AND.

Figure 8.3 illustrates the noisy gates BN for the unplugged CAT case study updated to include supplementary skills.

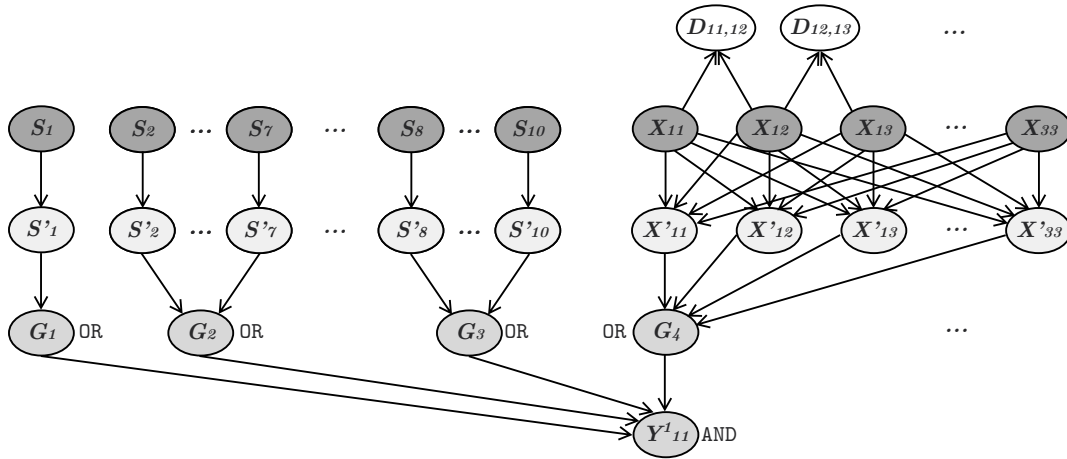


Figure 8.3 – Example of a constrained noisy gates BN modelling the unplugged CAT assessment rubric including supplementary skills.

8.3 Parameters' elicitation

Once the structure of the model is established for the CAT activity, it is necessary to set the values of the prior probabilities π_{*} , and the 12 inhibitors λ_{*}^t , $t = 1, \dots, 12$, for both the target and supplementary skills. Uniform prior probabilities, i.e., $\pi_{rc} = 0.50$, have been assigned to each skill. However, when conditioning given the constraints nodes $D_{rc,r'c'} = 1$, their probabilities, before the observation of any answer node, change. For example, for the unplugged CAT, they become $\pi_{11} = 0.95$, $\pi_{12} = 0.8$, $\pi_{13} = 0.5$, $\pi_{21} = 0.8$, $\pi_{22} = 0.5$, $\pi_{23} = 0.2$, $\pi_{31} = 0.5$, $\pi_{32} = 0.2$, $\pi_{33} = 0.05$. For the inhibition parameters, we compare two models: the *baseline* model, hereafter referred

to as Model B, where all inhibitors are set to the same value, and the *enhanced* model, hereafter referred to as Model E, with parameters elicited by a domain expert.

Model B may look trivial and unrealistic, but it allows one to understand better the effect of the constraints resulting from ordering the skills and supplementary skills on the model inferences. The constant value of λ was chosen equal to 0.2, except for the leak node, associated with all answer nodes and modelling a guess probability of 0.1, resulting in $\lambda_{\text{leak}} = 0.9$.

Model E builds on the baseline model to address the progressive complexity of the 12 tasks and the challenges students may encounter applying their skills to different schemas. The expert elicitation process involved grouping the 12 schemas into eight categories of increasing difficulties based on their characteristics: (i) T1, (ii) T2, (iii) T3, T4, (iv) T5, T6, (v) T7, T8, T9, (vi) T10, (vii) T11, (viii) T12. The expert assumed all tasks could be solved with 0D, 1D, and 2D algorithms. Moreover, given a schema t and a manifest variable Y_{rc}^t , the same inhibition probability was assumed for all relevant skills, meaning that all have the same probability of successfully being applied in solving schema t with level rc . In the proposed method, the inhibitor parameter λ_{rc} is used to model the probability of failing a task of a particular difficulty level rc , assuming the student has the necessary skills to solve the task. When a task is more complex or less help is available to the student, the value of rc increases, which means that the inhibitor parameter also increases. This is because when the student possesses the necessary skills to solve a difficult task, the probability of failing is higher than for a simpler task. Similarly, the inhibitor parameter λ_{rc}^t is assigned to a particular schema t and is used to model the difficulty of implementing a solution of level rc for that schema. A high value of λ_{rc}^t means that it is difficult to implement a solution of level rc for that schema. In other words, the inhibitor parameter λ_{rc}^t provides a measure of the difficulty of implementing a particular solution for a given schema at a given level of complexity.

While the students are generally expected to use 2D algorithms to solve the tasks optimally, there may be cases where a simpler 1D solution may be optimal. Nonetheless, in the current implementation, the first two tasks are designed to serve as starting points for students, introducing them to the activity. They are expected to be solved using simpler 1D algorithms. However, this particular case was not included in our model. This could have been described by setting high inhibitor values to indicate that certain 2D solutions are more difficult to implement than others, making them less likely to be chosen by students.

Our succinct elicitation setup allows for summarising both the BN topology and its parameter values graphically. An example, specifically referring to the unplugged CAT, is shown in the monochromatic rows at the bottom of Figure 8.4 and is explained in more detail in Figure 8.5 for schema T3.

The underlying BN has been implemented within the CREMA Java library [139], which supports the specifications of noisy gates and inference based on these parametric CPTs. The network size allowed for exact inferences using the Variable Elimination (VE) algorithm [52]. The model implementation is available on GitHub [6].

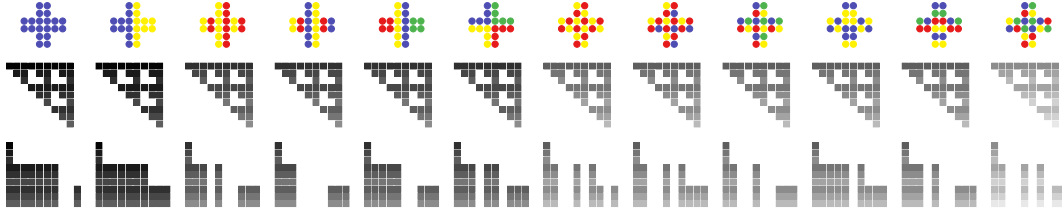


Figure 8.4 – The inhibition parameters for the unplugged CAT.

The 12 CAT schemas T (top); the values of the inhibition parameters λ_{rc}^t for the target skill nodes (centre); the value of the inhibition parameters λ_{Si}^t for the supplementary skill nodes (bottom). The inhibition parameters for both the target and supplementary skill are depicted as a matrix of nine rows representing the answers and as many columns as the number of modelled skills. The strength of the skill-answer relation has eleven levels, from 0.1 to 0.6, with a step of 0.05. Darker shades of grey mean lower skill-answer inhibition probabilities, and white squares denote non-relevant skills.

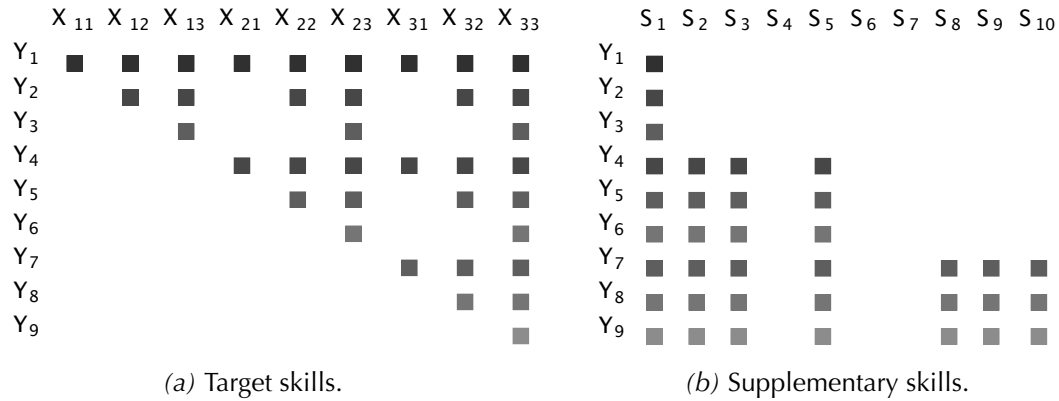


Figure 8.5 – The inhibition parameters for the unplugged CAT in schema T3.

Inhibition parameters λ_*^{T3} used in the ECS model for schema T3 (Zoom on schema T3 of Fig. 8.4). The parameters are divided into the target skills (top) and the supplementary ones (bottom). The supplementary skills S_4 (paint monochromatic squares), S_6 (paint monochromatic ls), and S_7 (paint monochromatic zigzags) are represented as empty columns because they cannot be used to solve task T3.

Part IV

Results

Chapter 9

Experimental study on the unplugged CAT

The content of this chapter has been adapted from the following article with permission of all co-authors and publishers:

- Piatti, A., **Adorni, G.**, El-Hamamsy, L., Negrini, L., Assaf, D., Gambardella, L., and Mondada, F. (2022). The CT-cube: A framework for the design and the assessment of computational thinking activities. *Computers in Human Behavior Reports* [227].

As an author of this publication, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation.

9.1 Summary

This chapter focuses on an experimental study using the unplugged CAT to assess the algorithmic skills of K-12 pupils, contributing to RQ4. As a preliminary study, our goal was to demonstrate how the CT-cube and the CAT can be used to evaluate students' algorithmic abilities. We explored how the task works in practice, analysing the algorithms produced by students and their performance according to the CT-cube dimensions, while considering the effects of age and gender. The chapter includes a detailed description of the study context, participant selection process, data collection methods, and the approach used for data analysis. Finally, the results are presented and discussed to provide insights into the effectiveness of the unplugged CAT for assessing algorithmic skills.

9.2 Methodology

9.2.1 Study context

The experimental study was conducted in Switzerland between March and April 2021. As the first study in our series, all testing took place in the canton of Ticino, part of Switzerland's Italian-speaking region, chosen for the ease and speed of identifying suitable classes and securing approval from local authorities through our established network. In our final study, we expanded the sample to other cantons, including regions with different spoken languages, to ensure broader geographical and linguistic representation (see Chapter 11).

9.2.2 Participant selection

To ensure a representative sample across all levels of compulsory education, with a balanced demographic of age, this study included eight classes from three public schools, in particular one preschool class (ages 3-6), three primary school classes (ages 6-11), and four lower secondary school classes (ages 12-16). It is important to note that the selection of schools and classes was not random. Rather, participating classes were selected through our network of contacts, and the schools were contacted and agreed to take part in the study.

Table 9.1 – Study participants (unplugged CAT).

Demographics analysis of participants by session, including school ID and type, HG, age category (mean and standard deviation), and gender distribution.

Session	School ID & type		HG	Age category		Female	Male	Total	
1	A	Preschool	0, 1, 2	3-6 yrs	(μ 4.9 \pm 0.9 yrs)	8	13	21	
2	B	Primary school	3	3-6 yrs	(μ 6.7 \pm 0.5 yrs)	4	8	12	
3	B	Primary school	5	7-9 yrs	(μ 8.7 \pm 0.6 yrs)	7	8	15	
4	B	Primary school	7	10-13 yrs	(μ 10.5 \pm 0.6 yrs)	8	11	19	
5	C	Lower secondary school	9	10-13 yrs	(μ 12.5 \pm 0.5 yrs)	8	7	15	
6	C	Lower secondary school	10	14-16 yrs	(μ 13.0 \pm 0.0 yrs)	5	2	7	
7	C	Lower secondary school	11	14-16 yrs	(μ 14.5 \pm 0.7 yrs)	9	5	14	
8	C	Lower secondary school	11	14-16 yrs	(μ 14.5 \pm 0.5 yrs)	2	4	6	
						(μ 9.9 \pm 3.5 yrs)	51	58	109

The study was conducted in multiple sessions, each assigned a unique ID corresponding to a specific activity or condition, allowing us to track and compare data across conditions and time points. All pupils in each class who had received parental consent were allowed to voluntarily participate in the study. For ethical considerations

regarding participant selection and informed consent procedures, refer to Section 1.5.4. A total of 109 pupils (51 girls and 58 boys) participated in the study. Table 9.1 provides a detailed breakdown of participant information across sessions, including school ID, HG, age and gender distribution.

9.2.3 Data collection approach and procedures

The study took place during regular class sessions and was administered in Italian. The procedure was designed to accommodate the participants' age and school environment. For younger students, task instructions were kept simple and engaging, while older students received more detailed guidance appropriate for their developmental level. The testing environment minimised distractions and ensured that the task did not interfere with the regular school schedule. No time limit was imposed to further reduce stress, allowing students to work at their own pace. To minimise interference with other students, two pupils at a time were randomly selected from the class and taken to a separate room to complete the task. Each pupil was randomly assigned to one of the available administrators from our research team, who explained the task and its goals, outlined the rules, and recorded session and participant details, as well as strategies employed by the pupils for each schema of the CAT on a protocol template (see Appendix C.1). For further details on the experimental setting, refer to Section 6.3.

The information recorded on the templates was later transferred to a database, which included session details, student information, and task performance data. Each session was assigned a unique identifier, with specific contextual information such as the date, canton, school ID and type, the HGs level, and the administrator's details. Student information was limited to the ID, gender, and date of birth, which was used to calculate their ages. The task and performance data recorded for each schema completed by the participants included details about the algorithm and interaction dimensions. Additionally, a separate database was created to store all the unique algorithms generated by schema. This included a description, the sequence of operations followed, the number of operations in the algorithm, the type of patterns used (e.g., point-by-point, rows, squares, etc.), and any redundancy present in the algorithms.

9.2.4 Data analysis approach

The data analysis aimed to provide insights into various aspects of the unplugged CAT assessment, focusing on participation, performance, algorithmic strategies, and the development of competencies across age groups and genders. Python was used for data processing, exploratory and statistical analyses, and visualisations of results [312].

Participation and performance

We begin by analysing the time-related performance metrics to gain insights into how long students took to complete the tasks. Next, we examine participation and success rates for each schema, exploring their distributions using descriptive statistics. Following this, we investigate the distribution of CAT scores across different age groups and performed a chi-square test of independence to assess whether there are significant relationships between age and performance outcomes [57, 195, 344]. We employ pairwise comparisons [344] to assess performance differences between age categories. Specifically, Tukey's Honestly Significant Difference (HSD) test was used to identify which specific age groups showed significant differences in their success rates [303]. To control for multiple comparisons and reduce the likelihood of Type I errors (false positives), we applied the Benjamini-Hochberg (BH) correction [30]. This adjustment modifies the p-values to control the false discovery rate, ensuring the reliability and accuracy of our findings. Additionally, we conduct a gender-based analysis to examine how performance varies between male and female students, focusing on the distribution of CAT scores. We assessed whether the variations in performance across genders were statistically significant.

Competencies development

In the second phase of analysis, we conduct a qualitative analysis on the development of algorithmic and interaction strategies across different age categories, exploring also on how these strategies changed within each schema. This analysis aimed to identify patterns in the approaches taken by students at various developmental stages. Additionally, we explored how these strategies varied by gender, providing insights into potential gender differences in task approach and competencies development.

Algorithms classification

Finally, we examined the diversity and frequency of algorithms generated by students for each schema, along with the distribution of algorithmic dimensions.

9.3 Results

9.3.1 Participation and performance

The time required to solve all the 12 schemas in the CAT varied significantly across participants, ranging from a minimum of 10 minutes for older pupils to a maximum of 45 minutes for the younger ones. This variation was primarily due to differences in

age and cognitive development. Older students tended to solve the tasks more quickly, demonstrating greater efficiency, while younger students required more time to understand and complete the schemas. The total time spent administering the CAT to all 109 students was approximately 36 hours.

In addition to time-related metrics, we analysed participation and success rates across schemas (see Table 9.2). Pupils tackled tasks up to schema 7 without intermissions. Beyond that point, some experiments were cut short due to time constraints, although the participation rate remained high, with a minimum of 95%. These interruptions were primarily due to the end of school hours or the limited attention span of very young preschool pupils [31, 178]. Regarding success rates, students were generally able to successfully complete all the tasks they were confronted with, demonstrating a strong engagement with the content and an ability to apply their algorithmic skills. However, more advanced schemas required additional time or support for younger students.

The high success rate across all schemas suggests that the tasks were accessible and well-aligned with students' cognitive capabilities. However, this also raises the possibility of a *ceiling effect*, where the tasks may have been too easy for some students, potentially limiting the ability to distinguish variations in skill levels. Future iterations could explore adjusting task complexity or introducing adaptive difficulty to better capture differences in AT.

Table 9.2 – Participation and success rates across schemas (unplugged CAT). Success rate is calculated from the number of students who attempted the schema.

	No. pupils participating (out of 109)	Participation (%)	Success (%)
Schema	1	109	100%
	2	109	100%
	3	109	100%
	4	109	100%
	5	109	100%
	6	109	100%
	7	109	100%
	8	107	98%
	9	105	96%
	10	105	96%
	11	105	96%
	12	104	95%

These results indicate the task suitability for pupils from preschool to secondary school, unlike most formal CT and AT assessments, which are limited to a pre-defined

age range. Indeed, the majority of studies focused on evaluating CT skills for a specific educational level, more often elementary or middle school. For example, the CTt by Román-González et al. [256] is designed for 7th and 8th-grade students (ages 12-14), although it can also be used to measure the CT from 5th to 10th grade (ages 10-16). The BCTt by Zapata-Cáceres et al. [348], which can be considered as an extension of the previous work, is mainly aimed at the first educational stages of primary school from the 1st to 4th grade (ages 5-10) and less adapted for middle school and beyond. Appropriate assessments of CT for younger children are generally more challenging and require considering developmental appropriateness in both format and content [241, 243]. Being an unplugged assessment, as shown in Section 6.3, the CAT does not require prior programming experience either, making it adequate for young pupils, as the CTt [256, 257], the BCTt [348] and TechCheck [243]. Moreover, avoiding confusing coding with CT skills is important.

We also investigated how performance differences could be attributed to age. Figure 9.1 shows that the distribution of CAT scores increases with age. This is attributable to the increased autonomy and the capability of using symbolic artefacts to describe the algorithms of older students. These differences are statistically significant, as confirmed by the chi-square test of independence ($\chi^2(12) = 270.2, p < 0.0001^{****}$).

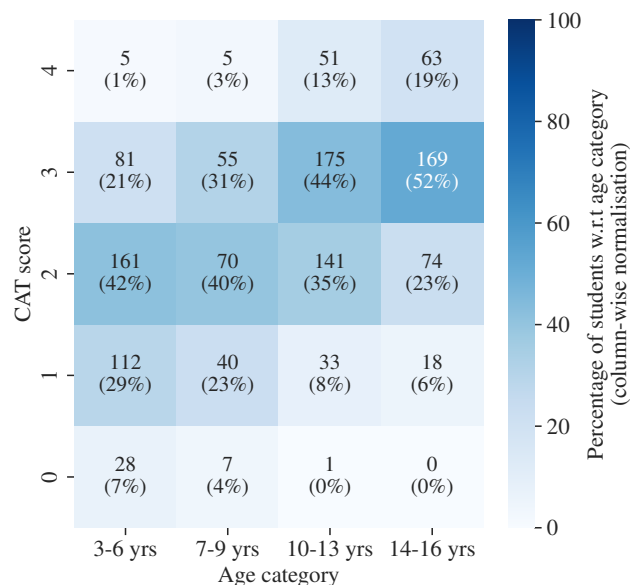


Figure 9.1 – Age-wise distribution of CAT score levels (unplugged CAT).

The pairwise comparison in Table 9.3 reveals the differences in CAT scores are statistically significant between all age categories, except for the two youngest groups

(ages 3-6 and 7-9). This leap can be attributed to the Swiss school system's organisation into educational cycles see Section 1.1.1), each with distinct objectives, with a key transition in the curriculum occurring after these two age categories [245, 246].

Table 9.3 – Pairwise comparison of CAT scores between age groups (unplugged CAT).

	3-6 yrs	7-9 yrs	10-13 yrs
7-9 yrs	$\chi^2(4) = 5,$ $p = 0.26$		
10-13 yrs	$\chi^2(4) = 75,$ $p < .0001^{****}$	$\chi^2(4) = 73,$ $p < .0001^{****}$	
14-16 yrs	$\chi^2(4) = 160,$ $p < .0001^{****}$	$\chi^2(4) = 182,$ $p < .0001^{****}$	$\chi^2(4) = 30,$ $p < .0001^{****}$

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, **** $p < 0.0001$

Turning our attention to performance variation by gender, Figure 9.2 illustrates the distribution of CAT scores across gender and age categories. While the overall improvement with age remains evident, no notable differences emerge between genders. This finding is supported by the results of the chi-square test of independence, which indicate no significant differences between genders for any age category ($p > 0.05$).

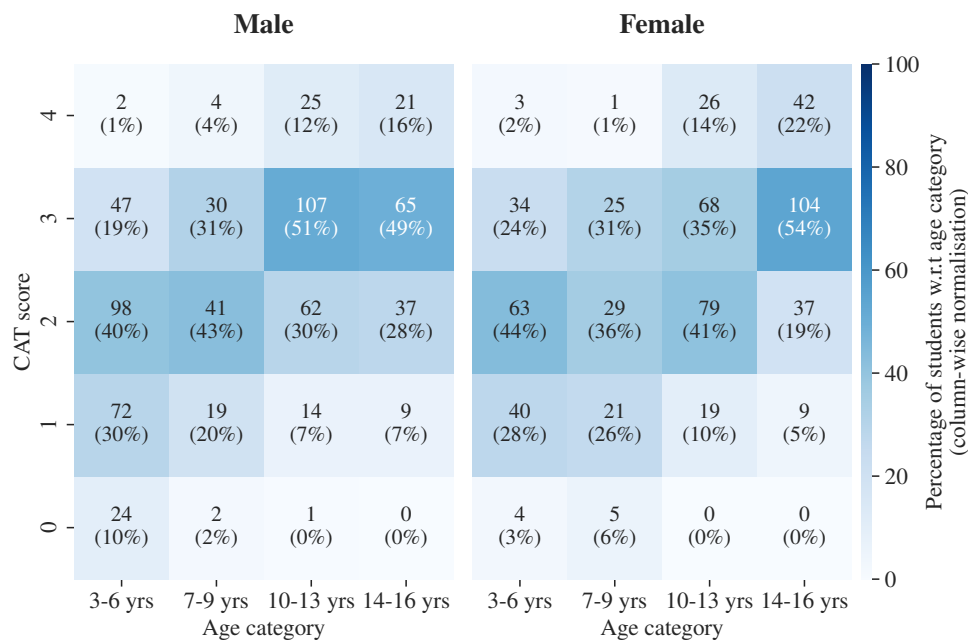


Figure 9.2 – Age-wise distribution of CAT score levels by gender (unplugged CAT).

9.3.2 Competencies development

To assess whether these results hold true across both algorithmic and interaction dimensions, we delve deeper into the strategies employed by pupils across different age categories, and subsequently, we analyse these strategies with respect to gender. Figure 9.3 illustrates the strategies employed by pupils to solve the CAT. Across all age categories, 1D algorithms are the most frequently used, while the proportion of 2D algorithms increases steadily with age. This trend suggests that pupils across all age groups are capable of creating complex algorithms, with their algorithmic skills demonstrating clear growth as they progress in age. From this visualisation, we can infer that

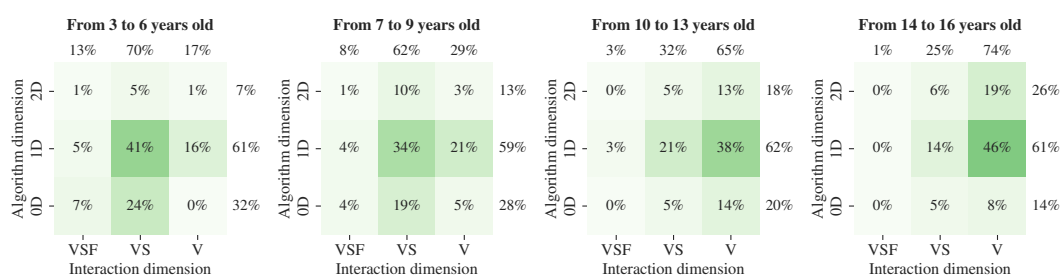


Figure 9.3 – Algorithmic and interaction strategies across age (unplugged CAT).

Percentages represent the proportion of each combination within their respective age groups. The sum of percentages across rows and columns reveals the aggregate preference or predominance for certain interaction-algorithmic strategies among different age groups.

the increase in CAT scores with age is primarily driven by an improved ability to utilise more complex artefacts (e.g., voice alone). The most frequent interaction dimensions in the first two age categories (ages 3-6 and 7-9) is VS, whereas for the last two age categories (ages 10-13 and 14-16) it is V. This trend suggests that older pupils are increasingly capable of handling more complex artefacts independently. Interestingly, younger pupils (ages 3-6) displayed non-autonomous behaviour (requiring visual feedback) in only 13% of cases. This percentage decreases progressively across age groups, reaching just 1% in the oldest group (ages 14-16).

The ability to conceive more complex algorithms also increases gradually with age. The two younger age groups (ages 3-6 and 7-9) are capable of conceiving 1D and 2D algorithms in many cases, and in some instances, they can even describe them solely through voice, demonstrating strong algorithmic skills from an early age. These results suggest that preschool pupils already exhibit algorithmic abilities, which can be elicited through the use of suitable artefacts, particularly embodied or iconic ones. This evidence is further supported by prior research demonstrating that working on CT skills with very young pupils is not only possible but also effective. Studies have

shown that CT skills can emerge and develop rapidly, even in preschool-aged children. For instance, Wohl et al. [339] achieved satisfactory learning outcomes with students under 5, concluding that unplugged activities can be used effectively to introduce computational concepts. Similarly, Dietz et al. [83] explored young children's developing capacities for problem decomposition and demonstrated that the skills necessary for this type of problem may be initiated to be fostered in preschool years.

A more detailed analysis of these strategies schema by schema (see Appendix C.3) shows that the improvement in interaction dimension is particularly evident from schemas S2 to S6. As for the algorithm dimension, the development of the algorithmic skills is more evident from schemas S7 to S9.

Turning again our attention to how these strategies vary by gender, Figure 9.4 confirms that the observations made for the full sample do not differ according to gender. As reported by Tikva and Tambouris [299], gender differences have been examined

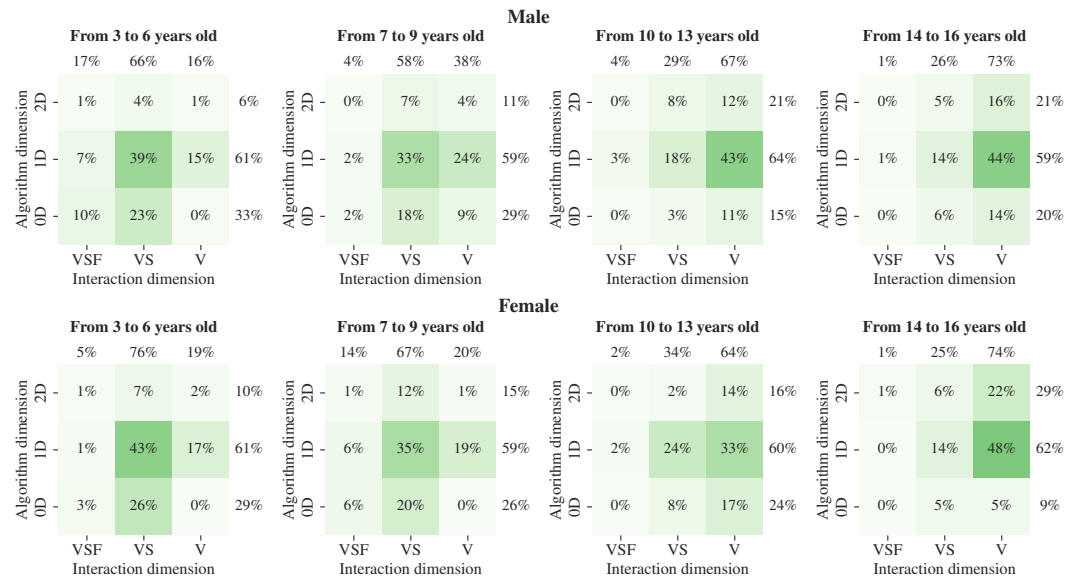


Figure 9.4 – Algorithmic and interaction strategies across age and gender (unplugged CAT).

in many studies, often leading to contradictory results. For example, Román-González et al. [256] reported a relationship between the role of gender and the development of CT. Other studies, such as Relkin et al. [243], Delal and Oner [76], Atmatzidou and Demetriadis [23] and Metin [190], verified gender relationship in their tests and concluded that no significant difference between genders appears when evaluating the students.

9.3.3 Algorithms classification

Finally, we focus on analysing the number of different algorithms generated by the students and the distribution of these algorithms according to their dimensions. Table 9.4 summarises, for each schema of the CAT, the number of different algorithms observed and the distribution of algorithms according to their dimension.

Overall, we have observed 137 different algorithms distributed across the 12 schemas of the CAT. For most schemas, there is a clear tendency for pupils to resolve

Table 9.4 – Algorithm dimensions distribution across schemas (unplugged CAT).
Values highlighted in bold indicate schemas where the algorithms produced with that algorithmic dimension are at least 20%.

		No. unique algorithms	0D (%)	1D (%)	2D (%)
Schema	1	5	2 %	95 %	4 %
	2	7	3 %	95 %	3 %
	3	10	2 %	84 %	14 %
	4	8	2 %	89 %	9 %
	5	5	0 %	100 %	0 %
	6	7	5 %	95 %	0 %
	7	19	50 %	18 %	31 %
	8	21	51 %	12 %	36 %
	9	9	52 %	0 %	48 %
	10	25	17 %	58 %	25 %
	11	15	14 %	61 %	25 %
	12	6	80 %	20 %	0 %
		137	23 %	61 %	16 %

the tasks by relying on algorithms of a certain dimension. Specifically, 1D algorithms are predominantly chosen in schemas S1 to S6, as well as in S10 and S11. In schemas S7 to S9, slightly more than half of the pupils opt for 0D algorithms, followed by 2D algorithms. In schema S12, the majority of pupils prefer using a 0D algorithm. Notably, there are a few schemas (S5, S6, and S12) where 2D algorithms are never selected, one case (S9) where 1D algorithms are never used, and one case (S5) where the point-by-point 0D algorithm is never used, suggesting that pupils preferred more efficient solutions with a higher algorithmic dimension.

Table 9.5 presents the frequency of algorithm usage across different schemas. Despite the variety of algorithms generated for each schema, a few algorithms are consistently recurrent, with two or three being the most commonly used. As previously mentioned, 1D algorithms are the most frequent, followed by 0D algorithms, while 2D

Chapter 10

Experimental study on the virtual CAT (pilot)

The content of this chapter has been adapted from the following articles with permission of all co-authors and publishers:

- **Adorni, G.** and Piatti, A. (unpublished). Designing the virtual CAT: A digital tool for algorithmic thinking assessment in compulsory education [10].
- **Adorni, G.**, Piatti, A., Bumbacher, E., Negrini, L., Mondada, F., Assaf, D., Mangili, F., and Gambardella, L. M. (2025). FADE-CTP: A Framework for the Analysis and Design of Educational Computational Thinking Problems. *Technology, Knowledge and Learning* [11].

As an author of these publications, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation, Supervision.

10.1 Summary

This chapter focuses on the pilot experimental study using the virtual CAT to evaluate the usability and suitability of this instrument for large-scale automated assessment of algorithmic skills in K-12 pupils, contributing to RQ4. By performing a similar analysis to that conducted with the unplugged CAT, we aim to obtain comparable results, demonstrating that the tool is usable and ready for broader data collection. The chapter includes a detailed description of the study context, participant selection process, data collection methods, and the approach used for data analysis. Finally, the preliminary results are presented and discussed, providing insights into the tool's effectiveness in assessing algorithmic skills and its potential for large-scale implementation. The code

used for the data analysis process is available on GitHub [4].

10.2 Methodology

10.2.1 Study context

The experimental study was conducted in Switzerland in March 2023. As this was a pilot study, during this testing phase, we remained within the canton of Ticino, which is part of the Italian-speaking region of Switzerland.

10.2.2 Participant selection

This study aimed to provide a preliminary selection of classes from different stages of compulsory education, focusing on the opposite ends of the system in Switzerland. The goal was to demonstrate the platform's effectiveness for diverse school types, with the intention of extending its applicability to the entire range of compulsory education. Thus, we included three classes from two public schools, in particular one preschool class (ages 3-6) and two secondary school classes (ages 10-13). It is important to note that the selection of schools and classes was not random. Rather, participating classes were selected through our network of contacts, and the schools were contacted and agreed to take part in the study.

As for the unplugged CAT, this study was conducted in multiple sessions. All pupils present in each class who had received parental consent were given the option to voluntarily participate in the study. For ethical considerations regarding participant selection and informed consent procedures, refer to Section 1.5.4. A total of 31 pupils (21 girls and 10 boys) participated in the pilot study. Table 10.1 provides a detailed breakdown of participant information across sessions, including school ID, HG, age and gender distribution.

Table 10.1 – Study participants (virtual CAT – pilot study).

Demographics analysis of participants by session, including school ID and type, HG, age category (age category (mean and standard deviation), and gender distribution.

Session	School ID & type		HG	Age category		Female	Male	Total	
1	X	Preschool	0, 1, 2	3-6 yrs	(μ 5.0 \pm 1.0 yrs)	3	4	7	
2	Y	Secondary school	8	10-13 yrs	(μ 11.2 \pm 0.6 yrs)	10	0	10	
3	Y	Secondary school	8	10-13 yrs	(μ 11.4 \pm 0.5 yrs)	8	6	14	
						(μ 9.9 \pm 2.8 yrs)	21	10	31

10.2.3 Data collection approach and procedures

The study took place during regular class sessions and was administered in Italian. It was structured into two main sequential modules: a training phase followed by a validation phase, both conducted on the same day, one after the other.

Recognised the importance of ensuring pupils' familiarity with the assessment tool, we designed and integrated a training module within the app that serves as a preparatory step for scholars, allowing them to become acquainted with the tool's interface and functionalities (see Figure D.2). During this phase, the administrator led a session guiding students through the app's features using 15 sample cross-array schemas for practice. This enables students to effectively navigate the tool, with teachers supporting the administrator and students as needed. Training sessions were held in groups based on device availability and typically lasted 30-45 minutes. The procedure was designed to accommodate the participants' age and school environment. For younger students, task instructions were kept simple and engaging, and they were guided to use only the CAT-GI, with simplified instructions and pacing to ensure comfort and understanding [122], while older students received more detailed guidance appropriate for their developmental level. No data collection occurred in this phase.

After completing the training, students move on to the validation session, during which data collection takes place. This module mirrors the original unplugged activity, where students solved 12 cross-array schemas. No time limit was imposed to ensure an optimal testing environment, allowing students to work at their own pace. For preschool pupils, to minimise distractions, two pupils at a time were randomly selected and taken to a separate room to complete the task.

To begin the validation process, session and student details are manually input into the app (see Figures D.4 and D.5). Each session was assigned a unique identifier, with specific contextual information such as the date, canton, school ID and type, the HGs level, and the administrator's details. Student information was limited to the ID, gender, and date of birth, which was used to calculate their ages. All information regarding tasks and performance is also collected for the virtual CAT. Additionally, this instrument allows to log timestamped actions performed by the students during the tasks, such as adding, confirming, removing, or reordering commands, updating command properties like colours or directions, resetting the algorithm, changing the mode of interaction or visibility, confirming task completion, or surrendering. All these data are compiled into a dataset, which has been made available through Zenodo for public access [2] after removing any potentially identifiable information (e.g., school and class) in alignment with open science practices in Switzerland [277].

10.2.4 Data analysis approach

The data analysis aimed to provide insights into various aspects of the virtual CAT assessment, focusing on participation, performance, algorithmic strategies, and the development of competencies across age groups. Python was used for data processing, exploratory analyses, and visualisations of results [312].

Participation and performance

We begin by analysing the time-related performance metrics to gain insights into how long students took to complete the tasks, comparing these across age categories and interaction dimensions. Next, we examine participation and success rates for each schema, exploring their distributions across age categories using descriptive statistics.

Competencies development

In the last phase of analysis, we conduct a preliminary qualitative analysis on the development of algorithmic and interaction strategies across age categories, aimed at identifying patterns in the approaches taken by students at various developmental stages.

10.3 Results

10.3.1 Participation and performance

Unlike the unplugged CAT, which required 36 hours for data collection with 109 participants, the virtual CAT enables simultaneous assessment of an entire class, provided each student has access to a device, allowing for assessments across multiple groups. The total time for the validation session with 31 students was approximately 4 hours (1 to 1.5 hours per session), a significant reduction compared to the unplugged version.

The time required to solve all 12 schemas shows notable differences across age categories (see Table 10.2). Younger students (ages 3-6) had slightly longer median times compared to older students (ages 10-13), but the interquartile ranges (Q1-Q3) indicate similar variability in completion times across both groups. This suggests that while age has some influence on task duration, the consistency of performance is comparable across age categories. The time students spent using each interaction dimension reveals notable differences (see Table 10.3). The use of the CAT-VPI (P) consistently required the most time, with a median duration of nearly 13 minutes. In contrast, the CAT-GI (G) had the shortest median duration, suggesting either a lower reliance on this dimension or quicker navigation. This aligns with expectations, as G is the less complex interaction dimension, making it more intuitive and easy for students to use.

Table 10.2 – Activity completion time across age categories (virtual CAT - pilot).

The time spent by students to complete all 12 schemas, including mean, minimum and maximum values, median, and interquartile ranges (Q1-Q3), grouped by age category. The aggregated statistics at the bottom summarise the overall completion times for the entire dataset.

		Mean	Min	Q1	Median	Q3	Max
Age category	3-6 yrs	15m 11s	09m 08s	10m 47s	12m 37s	17m 54s	26m 43s
	10-13 yrs	12m 48s	00m 10s	08m 12s	13m 00s	17m 16s	27m 48s

Table 10.3 – Time spent using each interaction dimension (virtual CAT - pilot).

The time students spent using a certain interaction dimension, including mean, minimum and maximum values, median, and interquartile ranges (Q1-Q3).

		Mean	Min	Q1	Median	Q3	Max
Interaction dimension	GF	06m 49s	00m 37s	03m 42s	05m 10s	08m 32s	26m 14s
	G	02m 24s	00m 10s	00m 52s	01m 13s	03m 21s	07m 45s
	PF	07m 09s	02m 10s	03m 25s	05m 27s	10m 05s	15m 58s
	P	13m 20s	00m 10s	09m 13s	12m 47s	17m 35s	27m 48s

The two tables reveal notable outliers in the time spent solving the 12 schemas. In Table 10.2, the maximum values significantly deviate from the upper quartiles. This suggests that while most students complete the tasks within a reasonable time frame, some struggle. This could be attributed to the complexity or unfamiliarity of the interaction dimensions rather than the age of the participants, as similar outliers are observed across both age groups. In fact, from Table 10.3, outliers are evident in the GF and P interaction dimensions, which represent the less and more complex modes of interaction, respectively. For instance, GF exceeds its upper quartile by approximately 18 minutes, which is more than 3 times, while P exceeds its third quartile by around 10 minutes, nearly double. These extreme values indicate that some students required disproportionately long durations, likely reflecting difficulties in navigating these dimensions or overreliance on them. While longer times with the P dimension are expected due to its complexity, the outlier in GF is more surprising. We can attribute this to the fact that less proficient students may gravitate toward this interaction mode, possibly because it appears simpler or more intuitive. In both tables, minimum values present notable outliers, with some students completing tasks in as little as 10 seconds. This suggests that some students might have rushed through the tasks or skipped steps, potentially indicating a lack of engagement. These outliers may also reflect brief interactions caused by technical issues, such as server disconnections

and data loss, which not only impacted the time but also affected task attempts and success rates. The presence of outliers underscores the need to consider individual differences in task completion times and interaction strategies when interpreting overall performance trends.

Table 10.4 provides an overview of participation and success rates across schemas for the two age categories. As expected from the previous analysis of completion times, not all pupils completed every schema. The reasons for this are multiple. Some experiments were cut short due to time constraints, as seen in the lower participation rates for the later schemas when students had to interrupt the activity. Additionally, there were some technical issues with the server, requiring some students to restart the full task, and some data was lost. Regarding success rates, older students generally per-

Table 10.4 – Participation and success rates across schemas and age categories (virtual CAT - pilot).

Success rate is calculated from the number of students who attempted the schema. Values highlighted in bold indicate schemas with success rate values exceeding 80%.

		Participation (%)			Success (%)		
		3-6 yrs (out of 7)	10-13 yrs (out of 24)	Total (out of 31)	3-6 yrs	10-13 yrs	Total (out of 31)
Schema	1	6 (86 %)	24 (100 %)	30 (97 %)	3 (50%)	22 (92%)	25 (83%)
	2	5 (71 %)	24 (100 %)	29 (94 %)	3 (60%)	21 (88%)	24 (83%)
	3	6 (86 %)	23 (96 %)	29 (94 %)	4 (67%)	19 (83%)	23 (79%)
	4	6 (86 %)	20 (83 %)	26 (84 %)	4 (67%)	17 (85%)	21 (81%)
	5	6 (86 %)	20 (83 %)	26 (84 %)	6 (100%)	16 (80%)	22 (85%)
	6	6 (86 %)	22 (92 %)	28 (90 %)	2 (33%)	21 (95%)	23 (82%)
	7	5 (71 %)	20 (83 %)	25 (81 %)	1 (20%)	12 (60%)	13 (52%)
	8	5 (71 %)	21 (88 %)	26 (84 %)	2 (40%)	18 (86%)	20 (77%)
	9	4 (57 %)	21 (88 %)	25 (81 %)	3 (75%)	20 (95%)	23 (92%)
	10	5 (71 %)	19 (79 %)	24 (77 %)	3 (60%)	18 (95%)	21 (88%)
	11	4 (57 %)	18 (75 %)	22 (71 %)	2 (50%)	14 (78%)	16 (73%)
	12	3 (53 %)	17 (71 %)	20 (65 %)	1 (33%)	14 (82%)	15 (75%)

formed better, while younger pupils showed more variation in their success rates. This could be attributed to differences in cognitive development, familiarity with digital tools, and task complexity, which likely posed more challenges for younger students. Overall, the success rate across both age groups is around 79%, indicating that pupils were able to engage effectively with the tasks. The schema with the lowest success rate appears to be S7, likely because it requires more advanced skills. Even the more challenging tasks, designed to be intentionally difficult, such as schemas S9 and S10, were successfully completed by many pupils (with success rates definitely higher than

80%). This demonstrates the platform’s potential to support learners across varying skill levels through its flexibility and diverse interaction modes.

10.3.2 Competencies development

To assess whether these results hold true across both algorithmic and interaction dimensions, we delve deeper into the strategies employed by pupils across different age categories, and subsequently, we analyse these strategies with respect to gender. Figure 10.1 illustrates the strategies employed by pupils to solve the CAT, while Appendix D.2 offers more detailed illustrations of the strategies schema by schema. Pupils

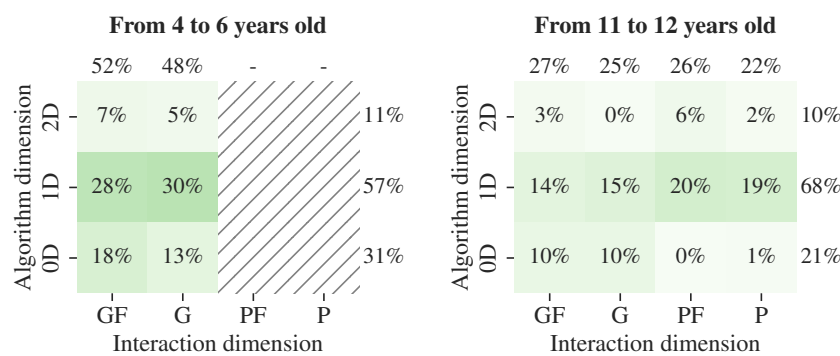


Figure 10.1 – Algorithmic and interaction strategies across age (virtual CAT - pilot).

Percentages represent the proportion of each interaction-algorithmic combination within each age group, with aggregated values shown across rows and columns. Since the younger age group could not use the CAT-VPI, only data for GF and G interactions are included.

across different age groups show a balanced use of interaction dimensions, highlighting the application’s capacity to cater to various interface preferences and levels of autonomy. Both groups are proficient in generating algorithms across all three algorithmic dimensions, with 1D algorithms being the most frequently used, consistent with the results of the unplugged CAT (see Section 9.3.2). While younger pupils consistently use all available interfaces without significant variation across algorithmic complexity levels, older pupils design algorithms of varying complexity based on the interaction type selected. Specifically, simpler algorithms are predominantly constructed using the CAT-GI, which offers an intuitive and cognitively light interaction style. In contrast, the CAT-VPI is favoured for more advanced tasks, as it supports better the creation of complex algorithms. This adaptability among older pupils highlights their strategic use of the platform’s features to address tasks of varying difficulty, emphasising the tool’s capability to support diverse skill levels and interaction strategies.

Chapter 11

Experimental study on the virtual CAT (main)

The content of this chapter has been adapted from the following article with permission of all co-authors and publishers:

- **Adorni, G.**, Artico, I., Piatti, A., Lutz, E., Gambardella, L. M., Negrini, L., Mondada, F., and Assaf, D. (2024). Development of algorithmic thinking skills in K-12 education: A comparative study of unplugged and digital assessment instruments. *Computers in Human Behavior Reports* [5].

As an author of this publication, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation, Supervision.

11.1 Summary

This chapter focuses on a main experimental study using the virtual CAT to assess the algorithmic skills of K-12 pupils in Switzerland, contributing to RQ4. We extended the preliminary analysis conducted with the unplugged CAT and the pilot study with the virtual CAT to provide more generalisable results from a larger sample. The chapter includes a detailed description of the study context, participant selection process, data collection methods, and the approach used for data analysis. Finally, the results are presented and discussed, providing insights into baseline competencies in AT in compulsory education, how these competencies develop across school grades, and how this development is influenced by instrumental factors such as the artefactual environment, as well as contextual factors like gender, educational environment (e.g., school level and grade), regional factors (e.g., canton of the school), and their interactions. The

code used for the data analysis process is available on GitHub [4].

11.2 Methodology

11.2.1 Study context

The experimental study was conducted between May and June 2023, specifically within the Solothurn and Ticino cantons, which are part of the German- and Italian-speaking regions of Switzerland, respectively.

11.2.2 Participant selection

To ensure a representative sample across all levels of compulsory education, with a balanced selection of age, gender and demographic origin, this study included nine classes across five public schools across Ticino and Solothurn cantons, in particular two preschool classes (ages 3-6), two primary school classes (ages 6-11), and five lower secondary school classes (ages 12-16). Participating classes were selected through our network of contacts, and the schools were contacted and agreed to take part in the study. All ethical considerations regarding participant selection and informed consent procedures are detailed in Section 1.5.4.

Table 11.1 – Study participants (virtual CAT - main).

Demographics analysis of participants by session, including canton (TI for Ticino and, SO for Solothurn), school ID and type, HG, age category (mean and standard deviation), and gender distribution.

Session	Canton	School ID & type		HG	Age category		Female	Male	Total	
1V	TI	A	Preschool	0, 1, 2	3-6 yrs (μ	5.0 ± 0.8 yrs)	6	7	13	
2V	SO	D	Preschool	2	3-6 yrs (μ	5.9 ± 0.3 yrs)	8	6	14	
3V	TI	E	Primary school	4	7-9 yrs (μ	7.7 ± 0.6 yrs)	7	8	15	
4V	SO	D	Primary school	6	7-9 yrs (μ	9.9 ± 0.3 yrs)	8	10	18	
5V	TI	F	Lower secondary school	8	10-13yrs (μ	11.6 ± 0.5 yrs)	11	9	20	
6V	TI	F	Lower secondary school	10	14-16yrs (μ	13.9 ± 0.8 yrs)	8	5	13	
7V	TI	G	Lower secondary school	10	14-16yrs (μ	13.6 ± 0.6 yrs)	7	7	14	
8V	TI	G	Lower secondary school	11	14-16yrs (μ	14.7 ± 0.5 yrs)	6	5	11	
9V	SO	D	Lower secondary school	11	14-16yrs (μ	15.5 ± 0.5 yrs)	4	7	11	
							(μ 10.7 \pm 3.6 yrs)	65	64	129

A total of 129 pupils (65 girls and 64 boys) participated in the study. Table 11.1 provides a detailed breakdown of participant information across sessions, including

canton, school ID, HG, age and gender distribution. While a balanced distribution is evident across factors like school type, HG, age category, and gender, there is a slight imbalance in geographic representation, with fewer students from Solothurn compared to Ticino, but overall, these demographic analyses offer valuable insights into the diverse characteristics of participants in both studies.

11.2.3 Data collection approach and procedures

The data collection procedures for this study closely mirrored those used in the virtual CAT pilot study (see Section 10.2.3). One notable difference in the main study was the variability in administering the activity across linguistic regions: one person managed sessions in the Italian-speaking region, while another conducted those in the German-speaking region. To ensure consistency, the Italian-speaking administrator supervised the later sessions in the German-speaking region. Differences in how the tutorials were administered could have impacted the delivery of instructional materials and the CAT's administration, potentially affecting the study's outcomes.

The validation session followed the same data collection procedure as the pilot study, with identical data being recorded. After removing any potentially identifiable information (e.g., school and class), the final dataset has been made publicly available through Zenodo [3], in alignment with open science practices in Switzerland [277].

11.2.4 Data analysis approach

The data analysis aimed to provide insights into various aspects of the virtual CAT assessment, focusing on participation, performance, algorithmic strategies, and the development of competencies across age groups and genders. Python was used for data processing, exploratory and visualisations of results [312], while R was used for statistical analyses [235].

Participation and performance

We begin by analysing the time-related performance metrics, first comparing task completion times across age categories, followed by an examination of these metrics across interaction dimensions. As part of this analysis, we examine the frequency of interaction dimensions to uncover preferences and tendencies in artefact use and autonomy among different age groups, as we hypothesise that a relationship exists between age categories and interaction dimensions. Next, we examine participation and success rates for each schema, exploring their distributions across age categories. Following, we investigated the distribution of CAT scores across different age groups and

performed a chi-square test of independence to assess whether there are significant relationships between age and performance outcomes [57, 195, 344]. Pairwise comparisons between age categories were conducted using Tukey's HSD test with BH correction [30, 303, 344].

Competencies development

In the second phase of analysis, we aimed to understand the influence of various factors on the algorithm dimension. This phase builds on observations and hypotheses made during the unplugged CAT experimental study and the pilot study with the virtual CAT, where potential correlations were identified but not statistically tested. To validate and extend these findings, we performed a series of statistical tests to assess the relationship between the complexity of the algorithms produced and factors such as age, interaction type and Trial and Error (T&E) approaches, as well as exploring the joint influence between some of these factors on AT.

To explore the direct relationship of algorithmic dimension with age and interaction dimension, we conducted an Analysis of Variance (ANOVA) to determine whether these factors significantly predicted the complexity of the algorithms produced [51, 57, 63, 124, 145]. Post-hoc analyses were performed using Tukey's HSD test with BH adjustments [30, 303, 344] and pairwise t-tests with Bonferroni correction [35, 221, 269] to explore these differences further. Finally, to assess the distribution of the higher algorithmic dimensions (2D) across different interaction dimensions and age categories, we performed a chi-squared test of proportions [57, 344]. Additionally, to further explore the relationship between interaction and algorithm dimensions, t-tests were conducted [26, 195, 333].

We also investigated students' T&E behaviours, specifically focusing on the predictors of task restarts. We used Ordinary Least Squares (OLS) regression to examine the relationship between restart frequency and factors such as schemas, age, gender, and interaction type [124, 145, 280, 349]. Subsequently, we analysed the impact of T&E behaviours on the algorithmic dimension to understand how this approach might influence the complexity of the algorithms produced and the achieved performance.

Finally, we explored the combined effect of factors such as interaction type, age, gender, and schemas on algorithmic dimension using ANOVA [51, 57, 63, 124, 145] and linear regression models [51, 96, 124, 184, 268]. We further investigated the relationship between interaction dimension and age by conducting an Estimated Marginal Means (EMMs) analysis to examine their combined effect on algorithm complexity [174]. Additionally, we used chi-squared tests of proportions to explore differences in the distribution of higher algorithm dimensions (2D) across various age categories and interaction types [57, 344].

11.3 Results

11.3.1 Participation and performance

Examining task completion times across age categories (see Table 11.2), we found that older students do not necessarily complete tasks faster, contrary to expectations. While median completion times generally increase with age, the second group deviates from this trend, suggesting a complex relationship between age and completion time. Interaction methods may also play a crucial role in this relationship. For example, older students might take longer to resolve tasks because they tend to use more advanced artefacts with higher autonomy, which are inherently more complex and could contribute to longer completion times.

Table 11.2 – Activity completion time across age categories (virtual CAT - main).

The time spent by students to complete all 12 schemas, including mean, minimum and maximum values, median, and interquartile ranges (Q1-Q3), grouped by age category. The overall summary statistics at the bottom provide an overview of completion times for the entire dataset.

		Mean	Min	Q1	Median	Q3	Max
Age category	3-6 yrs	20m 53s	03m 41s	15m 48s	21m 50s	27m 14s	35m 51s
	7-9 yrs	13m 13s	05m 27s	09m 52s	12m 24s	13m 54s	29m 39s
	10-13 yrs	26m 41s	05m 19s	17m 04s	25m 04s	41m 51s	52m 26s
	14-16 yrs	29m 34s	02m 51s	18m 58s	28m 08s	40m 29s	79m 36s
		23m 07s	02m 51s	12m 14s	20m 51s	30m 03s	79m 36s

Analysing students' interaction preferences among age categories, focusing on the least complex interaction dimension they achieved during the task and the most commonly used one, we aim to discern variations in usage patterns to contextualise our previous findings. The trend in Figure 11.1 reveals an evolution in interaction preferences as pupils grow older. Among the youngest group (ages 3-6), due to their limited exposure to technology, we restricted interactions to the CAT-GI interface. While these restrictions were lifted for the second group of students (ages 7-9), the CAT-GI remained the more popular assignment, even if some pupils began exploring the CAT-VPI, signalling an emerging interest in more complex interactions. Students in the third age group (ages 10-13) showed a balanced use of all four interaction dimensions, reflecting growing versatility and adaptability. Finally, older students (ages 14-16) predominantly opted for the most complex interaction dimensions (PF and P), demonstrating a preference for advanced and sophisticated methods. Our previously observed variations in task completion times across age groups can be explained by their interaction preferences. The second group (ages 7-9) showed the shortest completion times, likely

due to their continued reliance on simpler methods, like CAT-GI. In contrast, older students (ages 14-16) predominantly employed more advanced interactions, like CAT-VPI, which demand higher autonomy and cognitive effort, likely contributing to the longer completion times and supporting our previous hypothesis.

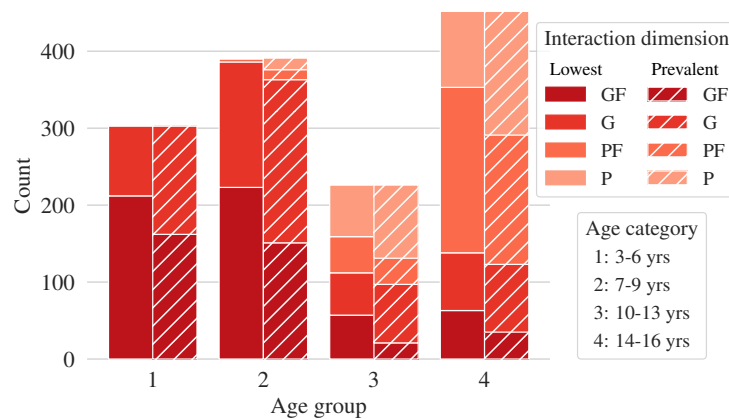


Figure 11.1 – Age-wise distribution of interaction dimensions (virtual CAT - main). The y-axis displays counts of the lowest and most used interaction dimensions, represented by solid and striped bars, across four age categories on the x-axis. Since the younger age group could not use the CAT-VPI, only data for GF and G interactions are included.

The analysis of time spent on specific interaction dimensions shows clear patterns (see Table 11.3). Users spend less time on the CAT-GI interface, aligning with pilot study findings (see Section 10.3.1), suggesting quicker navigation within this modality, while more time is spent on the CAT-VPI, with wide interquartile ranges (Q1-Q3) indicating variability in user engagement. Regardless of the interface used, users take longer to complete tasks when relying on visual feedback, likely due to the time needed for continuous monitoring and adjustment.

Table 11.3 – Time spent using each interaction dimension (virtual CAT - main). The time students spent using a certain interaction dimension, including mean, minimum and maximum values, median, and interquartile ranges (Q1-Q3).

		Mean	Min	Q1	Median	Q3	Max
Interaction dimension	GF	11m 38s	00m 29s	05m 21s	10m 00s	16m 59s	35m 36s
	G	02m 49s	00m 04s	00m 45s	01m 49s	04m 30s	11m 52s
	PF	21m 13s	02m 31s	08m 32s	18m 20s	30m 05s	79m 36s
	P	12m 15s	00m 01s	00m 29s	07m 18s	19m 33s	45m 20s

Table 11.4 provides a detailed overview of student participation and success on

individual schemas. Students were now able to skip tasks, an option unavailable in the unplugged CAT or the virtual CAT pilot. Despite this change, the participation rate remained notably high, reflecting strong determination and intrinsic interest, although a slight decline was observed in the later tasks. Regarding success rates, in the unplugged CAT, all students successfully completed each schema they attempted, as they were guided to correct any errors. However, success rates for the virtual CAT varied across schemas. Schema 1 had the highest success rate, reflecting its role as an introductory task, while Schema 12 showed lower success rates, indicating increased complexity. The drop in success rates for Schemas 2 and 8, despite a similar number of attempts as Schema 1, may suggest heightened task complexity or a mismatch between students' skills and schema demands. The non-linear decline in success rates as schema numbers increase suggests that students perceive varying levels of difficulty, which may not always align with educators' intended progression.

Table 11.4 – Participation and success rates across schemas (virtual CAT - main). Success rate is calculated from the number of students who attempted the schema. Values highlighted in bold indicate schemas with success rate values exceeding 80%.

	No. pupils participating (out of 129)	Participation (%)	No. pupils succeeding	Success (%)	
Schema	1	126	98%	119	94%
	2	127	98%	93	73%
	3	127	98%	100	79%
	4	129	100%	106	82%
	5	128	99%	109	85%
	6	127	98%	112	88%
	7	125	97%	98	78%
	8	126	98%	92	73%
	9	121	94%	98	81%
	10	118	91%	91	77%
	11	110	85%	82	75%
	12	110	85%	78	71%

Table 11.5 illustrates trends in student participation and success across age categories, specifically showing the percentage of students who completed the entire sequence of tasks or solved the entire sequence correctly. This type of participation rate is particularly beneficial for assessing whether the addition of the option to skip tasks has significantly affected participation rates for specific age groups. Overall, participation rates remain relatively high, with the second age group (ages 7-9) showing the highest participation. Regarding success rates, while we would expect an increase with age, this is not entirely the case, as the youngest pupils (ages 3-6) achieved the highest

success rate. This can likely be attributed to the relationship between age categories and the choice of interaction method, which may have influenced the outcomes.

Table 11.5 – Participation and success rates across age categories (virtual CAT - main). The number and percentage of students who attempted and successfully completed all 12 schemas, grouped by age category, along with the median and interquartile range (IQR). Success rate is calculated from the number of students participating.

		No. pupils participating in all schemas	Participation (%)	Median schemas participated (Q1-Q3)	No. pupils succeeding in all schemas	Success (%)	Median schemas succeeded (Q1-Q3)
Age category	3-6 yrs	18/27	67%	12 (11-12)	6	33%	10 (8-11)
	7-9 yrs	30/33	91%	12 (12-12)	4	13%	10 (9-11)
	10-13yrs	13/20	65%	12 (11-12)	3	23%	10 (7-11)
	14-16yrs	38/49	78%	12 (12-12)	11	29%	10 (8-11)

Examining the distribution of CAT scores across age groups, as seen for the unplugged CAT, we found that performance improves with age (see Figure 11.2), and these differences are statistically significant ($\chi^2 = 735.73$, $p < 1e - 15^{***}$). Again,

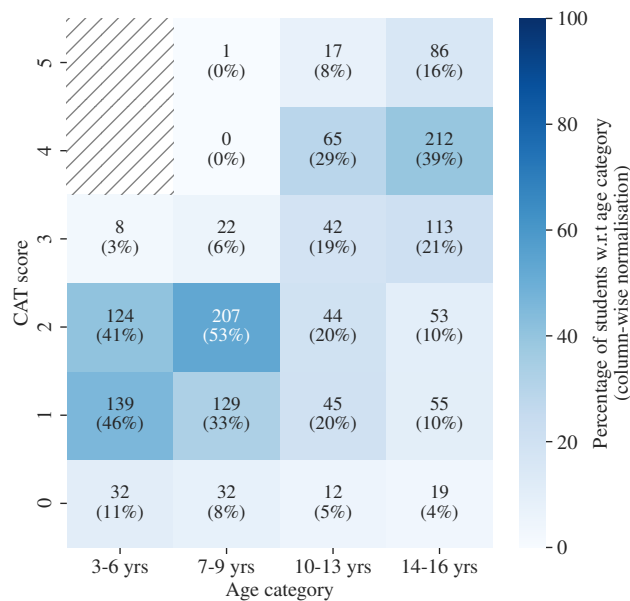


Figure 11.2 – Age-wise distribution of CAT score levels (virtual CAT - main).

Since the younger age group could not use the CAT-VPI, only data for GF and G interactions are included.

pairwise comparisons in Table 11.6 show that this significance holds for all age categories except the two youngest groups (ages 3-6 and 7-9).

Table 11.6 – Pairwise comparison of CAT scores between age groups (virtual CAT - main).

	3-6 yrs	7-9 yrs	10-13 yrs
7-9 yrs	$p = 0.542$		
10-13 yrs	$p < .0001^{****}$	$p < .0001^{****}$	
14-16 yrs	$p < .0001^{****}$	$p < .0001^{****}$	$p < .0001^{****}$

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, **** $p < 0.0001$

11.3.2 Competencies development

Impact of age on algorithmic skills

To understand the influence of age on the development of AT, we performed a series of statistical tests based on observations and hypotheses made during previous studies. ANOVA results show a positive correlation between age and algorithmic dimension ($p < 1e - 6^{****}$), indicating that as age increases, there is a corresponding increase in the complexity of the algorithm produced. Specifically, the older age category (ages 14-16) showed a higher algorithmic dimension and positive coefficients (0.23200, $1e - 06^{***}$). The chi-squared tests of proportions reveal significant age-related variations ($p < 0.0001^{****}$) in the higher algorithm dimensions (2D), with proportions increasing with age, reaching 36% for the older age category.

Impact of interaction strategies on algorithmic skills

Delving deeper into the strategies employed by pupils across different age categories, Figure 11.3 confirm that in the virtual domain, as for the unplugged, there is a developmental tendency towards more complex interactions, as shown in Figure 11.1. Regardless of age, there is a common tendency to create 1D algorithms, with a developmental progression towards more complex 2D algorithms, consistent with the unplugged study findings. The youngest age category (ages 3-6) demonstrated higher proficiency in conceiving complex 2D algorithms, surpassing simpler 0D ones, highlighting that virtual artefacts seemingly facilitate producing more complex algorithms at an earlier age.

We proceeded by conducting statistical analysis to determine whether the interaction dimension is a predictor of the algorithm dimension and identify if certain interaction strategies are more effective in producing complex algorithms than others.

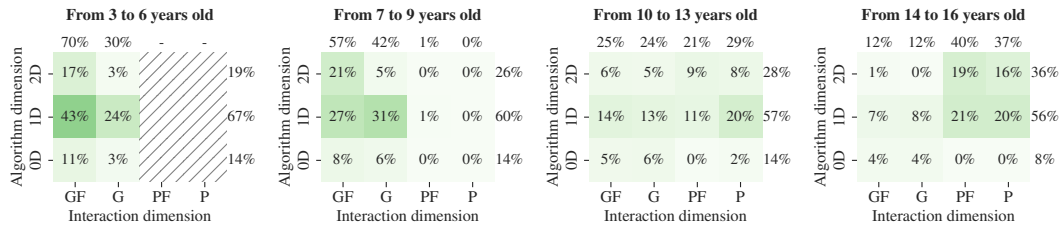


Figure 11.3 – Algorithmic and interaction strategies across age (virtual CAT - main). Percentages represent the proportion of each interaction-algorithmic combination within each age group, with aggregated values shown across rows and columns. Since the younger age group could not use the CAT-VPI, only data for GF and G interactions are included.

ANOVA tests showed that artefact dimension is a significant predictor of algorithmic dimension ($p < 1e - 15^{****}$). Further analysis using Tukey's HSD test with BH adjustment and pairwise t-tests with Bonferroni adjustment highlighted the substantial impact of the PF interaction on algorithm dimension when compared to other artefacts ($p < 0.001^{***}$). The chi-squared test of proportions confirmed these findings, showing significant variations in the production of higher algorithm dimensions (2D) across artefact categories ($\chi^2 = 140.38$, $p < 1e - 15^{****}$). The virtual PF artefact had the highest proportion of higher algorithm dimensions (46%), followed by P (39%), highlighting its effectiveness in promoting more complex algorithms. Finally, we performed T-tests to compare the algorithmic capabilities across the unplugged and virtual domains. The results reinforced our findings, revealing a significant difference in favour of the virtual interactions, which consistently led to the production of more complex algorithms ($t = -10.25365$, $p < 1e - 23$).

Impact of trial and error strategies on algorithmic skills

In this supplementary analysis, we aim to identify the predictors of students' use of the restart feature, a key aspect of T&E strategies, and assess its impact on the complexity of the algorithms produced. Our OLS regression analysis indicates that neither the characteristics of schemas (-0.0098 , $p = 0.161$) nor gender (-0.0059 , $p = 0.902$) significantly affect restart behaviour. Instead, a negative correlation was found between interaction dimension and restart frequency, indicating that students working with increasingly complex artefacts tend to restart their tasks less frequently. This is supported by Figure 11.4a, which shows that, for the simplest artefacts, the average number of restarts decreases as artefact complexity increases (-0.0283 , $p = 0.180$). Additionally, Figure 11.4b highlights a non-linear relationship between prevalent artefact complexity and restarts (-0.0168 , $p = 0.427$). The slight increase in restarts from non-autonomous (GF and PF) to autonomous (G and P) suggests that visual feedback

is important in supporting students' task participation and reducing restarts, possibly indicating that uncertainty arises when such feedback is absent. Also, age plays a significant role in restart behaviour, as reflected in a statistically significant inverse relationship (-0.0167 , $p = 0.012^*$), indicating that older students are less inclined to restart tasks. Figure 11.5 illustrates this decrease in the average number of restarts with increasing age categories. A peak in restarts among the second age group (ages 7-9) suggests increased exploration or developing problem-solving efficiency, while the decline in restarts among older groups (ages 10-13 and 14-16) may signal improved problem-solving skills and a greater ability to integrate past experiences, indicating a shift towards more advanced problem-solving approaches as students mature.

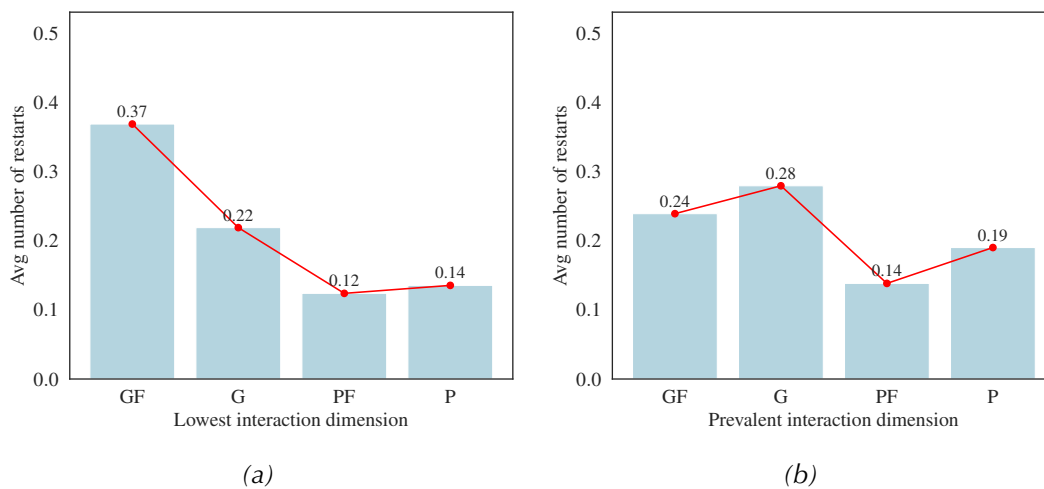


Figure 11.4 – Restarts distribution per interaction dimension.

Finally, our regression analysis examined the relationship between restart behaviour and the algorithm dimension. No significant association was found (-0.0221 , $p = 0.232$), and testing for non-linear effects using polynomial terms also revealed no significant patterns ($p = 0.200, 0.407$, and 0.497). This suggests that restart frequency does not predict algorithmic performance, either linearly or non-linearly. Regarding CAT scores, we obtained different results. While the direct link between restart frequency and CAT scores was not statistically significant (-0.0677 , $p = 0.110$), further polynomial terms investigation indicated a more complex relationship. Initially, higher restart frequency was associated with lower CAT scores (-0.4509 , $p = 0.001^{**}$), likely reflecting difficulty or confusion. However, the addition of a quadratic term showed a positive effect (0.0895 , $p = 0.031^*$), suggesting a non-linear pattern where, beyond a certain threshold, more restarts led to improved scores, suggesting a learning or optimisation process. The cubic term, however, was not significant (-0.0037 , $p = 0.151$),

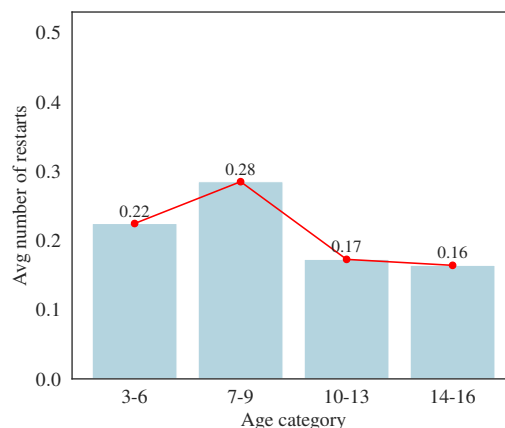


Figure 11.5 – Restarts distribution per age.

indicating no added explanatory value from further complexity.

This analysis reveals an intriguing dynamic between the variables. Age and interaction dimensions significantly influence restart frequency, indicating that restarts are not random but reflect user characteristics such as experience (linked to age) and the complexity of the interactions they navigate. Interestingly, restart frequency does not directly impact algorithm dimension, suggesting that the frequency of restarts is not a reliable indicator of algorithmic performance or the ability to handle complex problems. However, restart frequency does influence the CAT score, a composite measure of algorithm and interaction dimensions, through a non-linear relationship, highlighting the critical role of interaction dimension, which directly impacts the CAT score and indirectly shapes it through restart frequency, emphasising the need to consider interaction complexity when interpreting user performance.

Joint impact of interaction strategies and age on algorithmic skills

Examining the combined effect of interaction strategies and age on algorithmic skills, we found this relationship to be statistically significant ($p = 0.000106^{***}$). The linear regression model, EMMs analysis, and the chi-squared test of proportions revealed notable differences in how various interactions influence the algorithm dimension across age groups. The results show significant variations in interaction-algorithmic combinations for all age groups ($p < 0.01^*$). Younger children (ages 3-6) predominantly benefit from GF interactions (24%, $EMM = 1.090$, $SE = 0.0394$), while older age groups display more diverse patterns, with PF ($EMM = 1.465$, $SE = 0.0391$) and P ($EMM = 1.431$, $SE = 0.0409$) interactions becoming more common. prevalent, particularly for pupils aged 14-16, where PF interactions account for 47% and P interactions

for 44%. This is especially evident in the older age group (ages 14-16), where the most pronounced disparities are observed ($\chi^2 = 81.434$, $p < 1e - 15^{***}$), with PF interactions accounting for 47% and P interactions for 44%.

Joint impact of interaction strategies and gender on algorithmic skills

Examining the combined effect of interaction strategies and gender on algorithmic skills, ANOVA revealed a marginally statistically significant relationship ($p = 0.0521$), while no significance was found in the unplugged domain ($p = 0.2655$). The linear regression analysis further confirms these findings, highlighting a significant relationship between the simplest virtual artefact G and gender (0.21394 , $p = 0.00642^{**}$), indicating a smaller negative impact of the G interaction on algorithmic dimension for males than females.

Joint impact of interaction strategies and schemas on algorithmic skills

Examining the combined effect of interaction strategies and schemas characteristics on algorithmic skills, ANOVA revealed a significant interaction between these factors in both domains, with a stronger effect in the virtual domain ($p = 1e - 13^{****}$) compared to the unplugged one ($p = 1e - 9^{****}$). For the virtual CAT, linear regression analysis identifies significant positive interactions between artefacts and specific schemas. Notably, artefact P combined with schemas 2, 3, 8, 9, 10, 11, and 12, and artefact PF with schemas 9, 10, 11, and 12, lead to a higher algorithmic dimension. In contrast, for the unplugged CAT, the effects, while still significant, are primarily negative. For instance, artefact VS combined with schemas 3, 4, 7, 8, and 12, as well as artefact PF with schemas 9, 10, 11, and 12, significantly reduce algorithmic complexity.

Joint impact of age and gender on algorithmic skills

Examining the combined effect of age and gender on algorithmic skills, ANOVA revealed a significant interaction between these factors for the virtual domain ($p = 0.000115^{***}$) and less pronounced for the unplugged domain ($p = 0.00364^{**}$). In the virtual domain, linear regression analysis shows that male participants aged 10 to 13 years exhibit lower algorithmic skills compared to females (-0.29045 , $p = 0.00641$), reflecting a reversal of the pattern observed in the unplugged environment, where males in the same age range show higher algorithmic skills (0.22245 , $p = 0.011^*$).

Chapter 12

Factors influencing CAT performance

The content of this chapter has been adapted from the following article with permission of all co-authors and publishers:

- **Adorni, G.**, Artico, I., Piatti, A., Lutz, E., Gambardella, L. M., Negrini, L., Mondada, F., and Assaf, D. (2024). Development of algorithmic thinking skills in K-12 education: A comparative study of unplugged and digital assessment instruments. *Computers in Human Behavior Reports* [5].

As an author of this publication, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation, Supervision.

12.1 Summary

This chapter focuses on analysing the factors affecting K-12 pupils' performance in both the unplugged and virtual CAT assessments using hierarchical statistical models, contributing to RQ4. It examines how fixed and random effects explain performance differences and discusses the model selection process and the significance of predictors across both domains. The code used for the data analysis process is available on GitHub [4].

12.2 Methodology

We employed Linear Mixed Models (LMMs) to examine factors influencing student performance for both the unplugged and virtual CAT assessments. This approach is suitable for hierarchical data, as students are nested within sessions, schools, and can-

tons [137, 238]. LMMs allowed us to account for both fixed effects (e.g., gender) and random effects (e.g., variability across sessions), providing insights into how different predictors influence CAT performance while appropriately handling the hierarchical correlations in data.

The model selection and refinement process was carried out using the virtual CAT dataset. We started by constructing a baseline model using the Restricted Maximum Likelihood (REML) approach, with Satterthwaite's approximation for degrees of freedom to account for unequal variances and sample sizes [26]. This adjustment ensures a more accurate test of significance for the predictors. The model was implemented using the `lmer` function from the `lmerTest` package in R [170, 235].

After defining the initial model, the model selection process was carried out using the Likelihood Ratio Test (LRT) via the `anova` method in the `lmerTest` package in R [26, 51, 170, 235]. This method allowed us to compare alternative model specifications to assess the contribution of each predictor to explaining student performance.

Following model selection, we applied the chosen model to the virtual and unplugged datasets and conducted a Type III ANOVA with Satterthwaite's method to evaluate the significance of different terms in explaining the variability in CAT scores across both domains. For the virtual dataset, we examined the effect of task completion time on performance, exploring its relationship with variables such as age and interaction types. This analysis was performed solely for the virtual domain due to the lack of task completion data in the unplugged dataset. Task completion time was treated as a random effect to account for individual differences in completion pace, task complexity, and concentration, thereby avoiding potential biases that could arise if it were treated as a fixed effect.

Lastly, we combined the two datasets and introduced new fixed-effect predictors to differentiate between the virtual and unplugged domains. We then used LRT to assess the impact of gender on performance while controlling for school-related variability.

12.3 Results

12.3.1 Model selection and refinement

Our baseline model (M0), defined in Equation (12.1), includes various components to account for factors influencing student performance. The outcome variable `CAT_SCORE` represents the performance score. As fixed effects, we included `CANTON` due to the limited number of cantons (only two cantons) and `GENDER`, a binary predictor for gender. As random effects, we included `STUDENT`, accounting for natural heterogeneity among students, acknowledging unique factors like abilities, prior knowl-

edge, and unobserved characteristics inherent to each student; SESSION_GRADE captures variations related to HGs and testing sessions, such as time of day, classroom conditions, and differences introduced by various teachers, as well as disparities across educational levels, acknowledging each grade's unique curricular and teaching aspects; SCHOOL represents variability among different schools, encompassing their unique environments and resources; and SCHEMA accounts for variability among the 12 distinct tasks, isolating the task-specific characteristics. β_0 , β_1 , and β_2 are the coefficients for the fixed effects, while ϵ is the error term representing the unexplained variability.

$$\begin{aligned} \text{CAT_SCORE} = & \beta_0 + \beta_1 \cdot \text{CANTON} + \beta_2 \cdot \text{GENDER} + \\ & + u_{\text{STUDENT}} + u_{\text{SESSION_GRADE}} + \\ & + u_{\text{SCHOOL}} + u_{\text{SCHEMA}} + \epsilon. \end{aligned} \quad (12.1)$$

The baseline model has been analysed to assess the significance of the predictors, and as shown in Table 12.1, no statistically significant differences in CAT scores were found between students from Ticino and Solothurn cantons ($p = 0.719$). Similarly, there were non-significant variations in CAT scores between male and female students, with a negative coefficient suggesting slightly lower performance in males ($p = 0.497$). Variance estimates reveal variation in student performance across different grouping levels, with notable differences observed at the school and student levels.

To further refine our analysis, we performed an LRT to evaluate the contribution of canton and gender as predictors. The results in Table 12.2 indicate that including the canton predictor doesn't significantly improve the model's fit ($p = 0.5998$), likely due to the limited representation of Swiss cantons in the data (only 2 out of 26 sampled). Therefore, we opt for the simpler reduced model (M1).

To assess the impact of gender, whose effect was not statistically significant in the baseline model (M0), we compared three models, each considering different combinations of predictors: the reduced model (M1) without the canton predictor, another reduced model (M2) without both canton and gender and an improved model (M3) without canton but with gender as a random slope within schools. The assumption in model M3 stems from observations in our experimental studies, which suggest that contextual factors, such as school environment and resources, may influence gender dynamics. The LRT results in Table 12.3 show that including gender as a fixed effect (M2) does not significantly enhance model fit ($p = 0.4881$), suggesting that gender alone is not a substantial predictor of CAT scores. However, the improved model (M3) showed a significant improvement in fit ($p < 1e-3^{***}$), indicating that gender-related differences may vary across schools.

Table 12.1 – Baseline model (M0) summary.

The REML criterion at convergence is 4044.3 for the baseline linear mixed-effects model.

(a) Scaled residuals.

Min	Q1	Median	Q3	Max
−3.785	−0.582	0.050	0.611	3.569

(b) Random effects.

Groups*	Name	Variance	SD ^a
Student	(Intercept)	0.490	0.700
Schema	(Intercept)	0.057	0.238
Session-Grade	(Intercept)	0.054	0.232
School	(Intercept)	0.931	0.965
Residual		0.756	0.870

^a Standard deviation

* Student (129), Schema (12), Session-Grade (9), School (5);
Number of observations: 1457

(c) Fixed effects.

	Estimate	SE ^b	df	t ^c	p ^d
(Intercept)	2.169	0.986	2.826	2.200	0.121
Gender ¹	−0.090	0.132	120.082	−0.681	0.497
Canton ²	0.439	1.101	2.815	0.398	0.719

^b Standard error

^c t value

^d p value = $Pr(> |t|)$

¹ Gender: Male

² Canton: Ticino

Table 12.2 – LRT to evaluate the inclusion of canton as a predictor.

Comparison between the reduced model (M1) without the canton predictor and the baseline model (M0) including it.

Model	AIC ^a	χ^2	p ^b
M1	4058.1		
M0	4059.8	0.275	0.5998

^a Akaike Information Criterion for the model evaluated as
−2 · (logLik − npar). Smaller is better.

^b p value = $Pr(> \chi^2)$

Table 12.3 – LRT to evaluate the inclusion of gender as a predictor. Comparison between the reduced model (M1) without the canton predictor, another reduced model (M2) without canton and gender, and an improved model (M3) without canton, but that considers gender as a random slope within schools.

Model	AIC ^a	χ^2	p^b
M2	4056.6		
M1	4058.1	0.481	0.4881
M3	4048.3	11.785	0.0006***

^a Akaike Information Criterion for the model evaluated as $-2 \cdot (\log\text{Lik} - \text{npar})$. Smaller is better.

^b p value = $Pr(> \chi^2)$

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, **** $p < 0.0001$

Grey shadings indicate statistically significant fixed effects

12.3.2 Model application and predictor significance

Given these results, we consider model M3, defined in Equation (12.2). This model comprises the intercept coefficient β_0 , and the coefficients for the effect of gender within schools β_1 and β_2 . The random effects include STUDENT, SESSION_GRADE, and SCHEMA. Finally, ϵ_{GENDER} represents the error term for the interaction between gender and school, and ϵ denotes the unexplained variability in the model.

$$\begin{aligned} \text{CAT_SCORE} = & \beta_0 + (\beta_1 + \beta_2 \cdot \text{GENDER} + \epsilon_{\text{GENDER}} | \text{SCHOOL}) + \\ & + u_{\text{STUDENT}} + u_{\text{SESSION_GRADE}} + \\ & + u_{\text{SCHEMA}} + \epsilon. \end{aligned} \quad (12.2)$$

Once the model was defined, it was applied to both the unplugged and virtual CAT data. The model summary in Table 12.4 shows a significant intercept in the fixed effects for both domains (U: $p = 0.014^*$, V: $p = 0.00286^{**}$), suggesting that the average CAT score significantly differs from zero when accounting for random effects, indicating a baseline proficiency level within the student population. Exploring random effects, school-level differences account for the largest variance in both domains (U = 0.47939, V = 0.87332), suggesting that school-specific factors strongly influence CAT scores. Gender variance is similar across domains, with a lower positive correlation in the unplugged domain (0.14) and a negative one in the virtual domain (-0.49), suggesting different gender dynamics: in the virtual domain, schools with higher overall performance tend to have lower scores for male students, and vice versa. The virtual domain has higher student variance (U = 0.23476, V = 0.40687), indicating greater individual variability, possibly influenced also by unmeasured factors. Session-grade variance is smaller for unplugged (U = 0.02378, V = 0.05883), indicating more consistent performance, likely due to stable session-grade factors like age progression, curriculum complexity, teaching methods, or cohort effects. Lastly, the unplugged domain shows a larger variance in schemas compared to the virtual domain (U = 0.19660, V = 0.05605), suggesting that factors such as the nature of the activity, the way information is presented, and features like the ability to skip or solve schemas in a preferred order contribute to this variation.

The ANOVA results in Table 12.5 further confirm the significance of the random effects, particularly for schema and student ($p < 1e - 15^{***}$), showing that task type and individual student differences significantly affect CAT scores. Interestingly, the impact of gender on CAT scores varies significantly across schools in the virtual domain ($p = 0.002^{**}$) but not in the unplugged one ($p = 0.803$), suggesting that gender-related factors have a greater effect in virtual environments.

Table 12.4 – Model (M3) summary.

The REML criterion at convergence is 3377.3 for the unplugged domain (U) and 4032.5 for the virtual domain (V) in the linear mixed-effects models.

(a) Scaled residuals.

	Min	Q1	Median	Q3	Max
U	−3.574	−0.656	−0.048	0.563	2.855
V	−3.797	−0.578	0.048	0.616	3.537

(b) Random effects.

	Groups*	Name	Variance	SD ^a	Corr
U	Student	(Intercept)	0.235	0.485	
	Schema	(Intercept)	0.197	0.443	
	Session-Grade	(Intercept)	0.024	0.154	
	School	(Intercept)	0.479	0.692	
		Gender ¹	0.027	0.164	0.14
	Residual		0.671	0.819	
V	Student	(Intercept)	0.407	0.638	
	Schema	(Intercept)	0.057	0.237	
	Session-Grade	(Intercept)	0.059	0.243	
	School	(Intercept)	0.873	0.935	
		Gender ¹	0.253	0.503	−0.49
	Residual		0.756	0.870	

^a Standard deviation

¹ Gender: Male

* U: Student (109), Schema (12), Session-Grade (8), School (3);

Number of observations: 1280

V: Student (129), Schema (12), Session-Grade (9), School (5);

Number of observations: 1457

(c) Fixed effects.

		Estimate	SE ^b	df	t ^c	p ^d
U	(Intercept)	2.827	0.432	2.397	6.54	0.014*
V	(Intercept)	2.429	0.389	4.194	6.244	0.003**

^b Standard error

^c t value

^d p value = $Pr(> |t|)$

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, **** $p < 0.0001$

Grey shadings indicate statistically significant fixed effects

Table 12.5 – Type III ANOVA, with Satterthwaite’s method, on model (M3). Rows shaded in grey indicate statistically significant variables.

		AIC ^a	LRT ^b	p ^c
U		3393.3		
	Schema	3653.2	261.903	< 1e – 15****
	Gender ^d	3389.7	0.438	0.803
	Session-Grade	3393.1	1.780	0.182
	Student	3584.7	193.418	< 1e – 15****
V		4048.5		
	Schema	4112.5	65.75	< 1e – 15****
	Gender ^d	4056.7	12.20	0.002**
	Session-Grade	4049.6	3.11	0.078
	Student	4397.0	350.52	< 1e – 15****

^a Akaike Information Criterion for the model evaluated as $-2 \cdot (\log\text{Lik} - \text{npar})$. Smaller is better.

^b LRT statistic; twice the difference in log-likelihood, which is asymptotically chi-square distributed.

^c p value = $Pr(> \chi^2)$

^d Gender in (1 + Gender | School)

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, **** $p < 0.0001$

Grey shadings indicate statistically significant fixed effects

Gender influence within schools on performance

Comparing gender-related school performance in the two domains, Figure 12.1 shows no significant effects for the unplugged domain, while the virtual domain exhibits variability in CAT scores between male and female students across schools. Certain schools (e.g., D) show higher CAT scores for male students than others (e.g., F), challenging the idea of a uniform gender effect.

Focusing on average performances across schools, we discern differences in the baseline performance for both domains. In the unplugged case, A performs below average, B slightly below, and C notably above. In the virtual case, schools exhibit varied impacts on female students, with some (A, D, and E) showing decreased CAT scores and others (F and G) demonstrating increased scores.

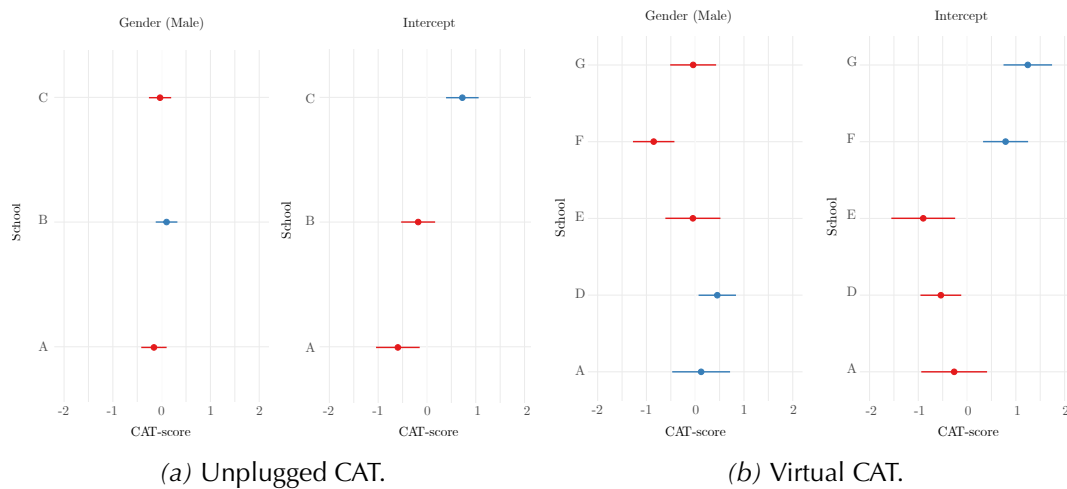


Figure 12.1 – Gender-related school performance variations.

The plot on the left captures the variability in CAT scores between schools for male students compared to female students, illustrating how scores differ across schools based on gender. On the right, the plot shows the intercept, representing the average variability between schools, exclusively focusing on female students. Blue points represent scores above average, while reds those below. Horizontal lines represent the estimates' confidence intervals.

Individual student variability in performance

The analysis of student performance in Figure 12.2 shows significant individual variability in both domains. The presence of high achievers (blue dots to the right) and those facing challenges (red dots to the left) is consistent across both datasets, highlighting a diverse range of performances. This indicates a substantial amount of unexplained variability not accounted for by other factors considered in the study.

Despite accounting for school-level differences in the model, unexplained variability in CAT scores persists among students, indicating significant differences in performance residuals across various schools. This suggests that factors associated with the distinct educational environments of each school might contribute to the observed variance. Statistical analysis, specifically Levene's test, supports this observation ($p < 1e - 15^{****}$).

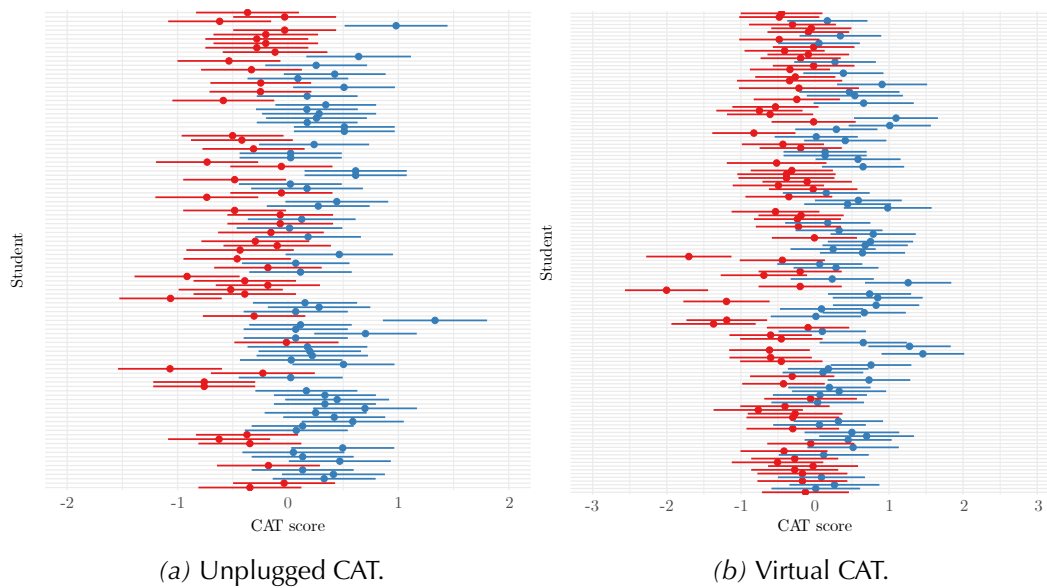


Figure 12.2 – Individual student performance variations.

Each point represents the student deviation from the average CAT score, with blue indicating scores above average and red below. Horizontal lines represent the estimates' confidence intervals.

Session grade impact on performance

Examining the impact of session and grade on scores in Figure 12.3, no statistical differences in performance across sessions and grades are observed in both the unplugged and virtual domains. The pattern of fluctuations implies a complex relationship between sessions, grades, and CAT scores. Notably, lower performance is observed from the initial to the middle sessions, coinciding with lower HarmoS grades (HGs). Positive deviations in higher sessions suggest that older students generally perform better. This consistency implies that advanced cognitive skills and better adaptation to educational demands may contribute to improved performance among older students.

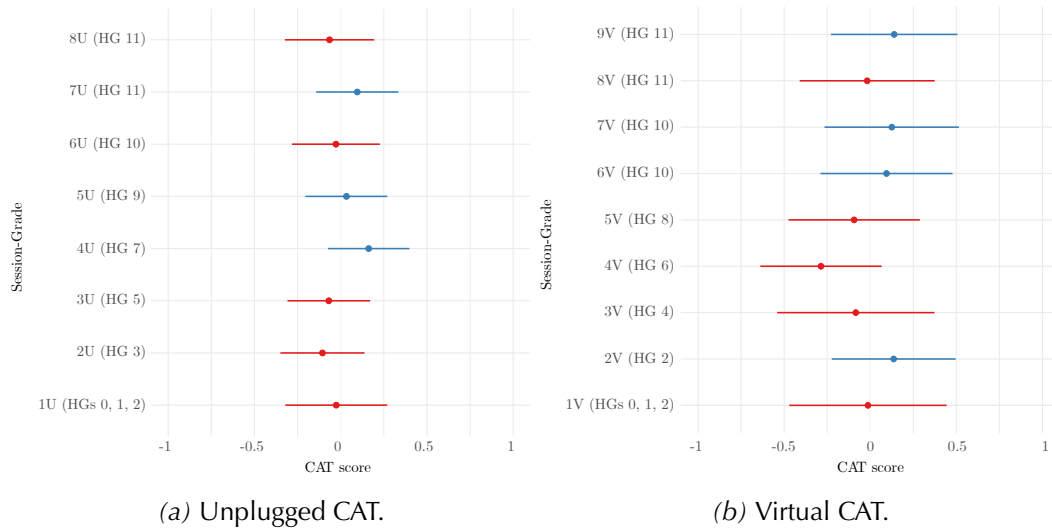


Figure 12.3 – Session-Grade performance variations.

Each point represents the session deviation from the average CAT score, with blue indicating scores above average and red below. Horizontal lines represent the estimates' confidence intervals.

Schema-based differences in performance

Task performance varies across different schemas in both the unplugged and virtual domains, as depicted in Figure 12.4. For the unplugged dataset, initial schemas (1 to 6) generally yield good performance, although there's a decline as the schema number increases, hinting at rising task difficulty. Schemas 7 to 9 show below-benchmark scores but with improving trends, suggesting student adaptation or better task alignment. Scores rise in schemas 10 and 11 but drop significantly in schema 12, possibly due to task difficulty or misalignment with student abilities. A consistent decreasing trend is observed in the virtual dataset, although with some irregularities. Performance is above the benchmark for less difficult tasks (1 to 5) and declines below the benchmark (6 to 12) with increasing task difficulty. Notably, the mean CAT score for schema 8 is above the benchmark, suggesting better-than-expected performance on average. The non-linear decline in performance as schema numbers increase suggests that students perceive varying levels of difficulty, which may not align with the intended task progression.

To explore performance trends and irregularities across different tasks, we specifically looked at the algorithm dimension instead of overall performance. This examination pertains specifically to the virtual CAT, where we have precise and comprehensive information on all the commands students use in crafting their algorithms.

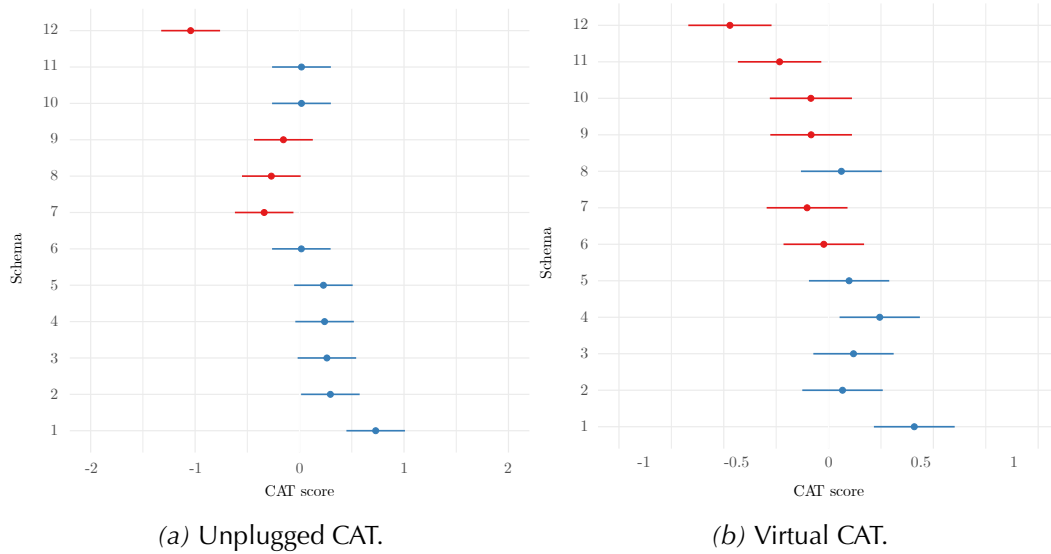


Figure 12.4 – Schema-based performance variations.

Each point represents the schema deviation from the average CAT score, with blue indicating scores above average and red below. Horizontal lines represent the estimates' confidence intervals.

Figure 12.5a indicates that the algorithm dimension varies across tasks, suggesting that students adapt their problem-solving strategies to each task rather than following a linear regression of algorithm complexity. Notably, for schemas 1, 2, 5, 6, and 12, students often use 1D dimensional algorithms driven by practical considerations.

Sometimes, a simpler, less complex algorithm with fewer moves is more effective than a more intricate one. This preference for efficiency doesn't imply lower performance but reflects a pragmatic approach to problem-solving, as argued in Section 7.4.2. To assess student performance, we introduced an alternative method considering both algorithm complexity and efficiency. The adapted algorithm dimension metric, presented in Figure 12.5b, demonstrates a more linear decrease in average algorithm dimensions. Figure 12.6b shows the original and updated distribution of performance across schemas using the new metric.

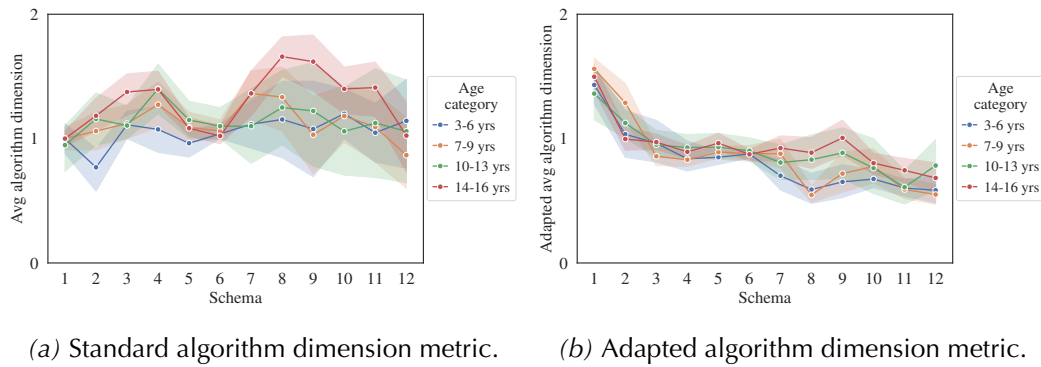


Figure 12.5 – Algorithm dimension variations across age categories at schema level. The y-axis represents the average variations in algorithm dimension for each age category, plotted against different schemas on the x-axis.

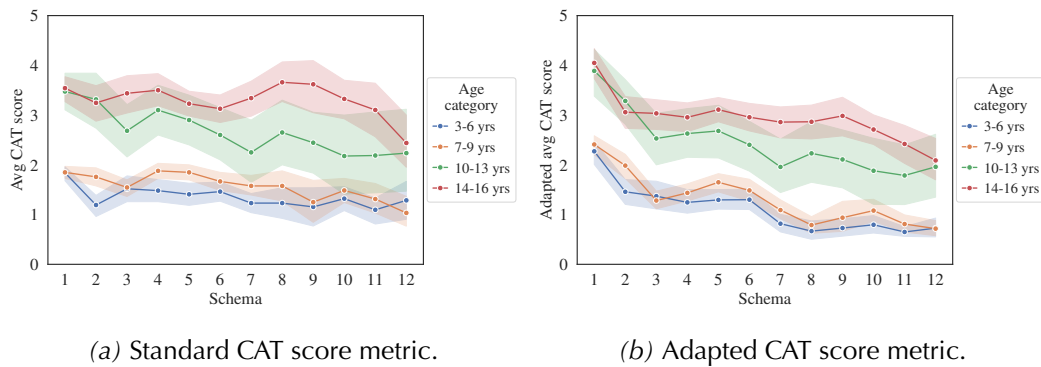


Figure 12.6 – Performance variations across age categories at schema level. The y-axis represents the average variations in CAT score for each age category, plotted against different schemas on the x-axis.

Task completion time effects

In the concluding phase of our analysis, we incorporated task completion time as a random effect into our existing model. This specific examination was exclusive to the virtual CAT, benefiting from detailed records of task completion times. We aimed to uncover the correlation between the time students spent on tasks and their ensuing performance levels. Figure 12.7 shows a non-linear relationship between task completion time and performance. Both extremely brief and significantly extended durations appear beneficial, resulting in higher CAT scores. This observation suggests that rapid responses may be driven by strong intuition or familiarity, while longer times may reflect a more analytical approach, likely enhancing performance. On the other hand, intermediate completion times do not seem to capitalise on the strengths of either ap-

proach, potentially explaining the observed dip in scores and the negative impact on performance associated with moderate haste.

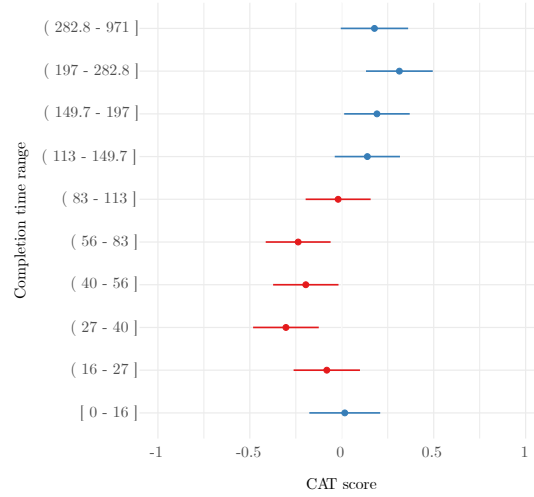


Figure 12.7 – Task completion time and performance variations.

Each point represents the deviation of task completion time intervals from the average CAT score, with blue indicating scores above average and red below. Horizontal lines represent the estimates' confidence intervals.

12.3.3 Domain-specific analysis

To understand the factors influencing student performance in both unplugged and virtual settings, we combined the datasets to formulate the final model (M4), defined in Equation (12.3), aimed at assessing how the domain impacts CAT scores, along with other contributing factors. The difference from M3 is the inclusion of the variable domain as a predictor of the CAT score.

$$\begin{aligned}
 \text{CAT_SCORE} = & \beta_0 + \beta_1 \cdot \text{DOMAIN} + \\
 & + (\beta_2 + \beta_3 \cdot \text{GENDER} + \epsilon_{\text{GENDER}|\text{SCHOOL}}) + \\
 & + u_{\text{STUDENT}} + u_{\text{SESSION_GRADE}} + \\
 & + u_{\text{SCHEMA}} + \epsilon.
 \end{aligned} \tag{12.3}$$

From Table 12.6, we observe that variations in student performance across different groups, including individual students, session grades, schemas, and gender across schools, align with patterns identified in the model (M3) on individual datasets. Nevertheless, the model revealed that the domain effect on CAT scores lacked statistical significance ($p = 0.901$), thereby strengthening the coherence of these results across

Table 12.6 – Model (M4) summary.

The REML criterion at convergence is 7445.5 in the linear mixed-effects model.

(a) Scaled residuals.

Min	Q1	Median	Q3	Max
-4.005	-0.628	0.005	0.618	3.640

(b) Random effects.

Groups*	Name	Variance	SD ^a	Corr
Student	(Intercept)	0.328	0.573	
Session-Grade	(Intercept)	0.039	0.196	
Schema	(Intercept)	0.110	0.331	
School	(Intercept)	0.758	0.871	
	Gender ¹	0.167	0.409	-0.39
Residual		0.730	0.854	

^a Standard deviation

¹ Gender: Male

* Student (238), Session-Grade (17), Schema (12), School (7);
Number of observations: 2746

(c) Fixed effects.

	Estimate	SE ^b	df	t ^c	p ^d
(Intercept)	2.638	0.382	10.088	6.905	<1e-04****
Domain ²	-0.040	0.317	12.219	-0.127	0.901

^b Standard error

^c t value

^d p value = $Pr(> |t|)$

² Domain: Virtual

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, **** $p < 0.0001$

Grey shadings indicate statistically significant fixed effects

various settings. This underscores the robustness of the conclusion, highlighting the significance of domain-independent factors in shaping CAT scores.

The ANOVA results in Table 12.7 highlight that all factors significantly influence CAT scores in both virtual and unplugged settings.

To assess the impact of gender on CAT scores, we compared three models, each considering different combinations of predictors: the model with gender as a random slope within schools (M4), a reduced model without gender as predictor (M5), and

Table 12.7 – Type III ANOVA, with Satterthwaite’s method, on model (M4).

	AIC ^a	LRT ^b	p ^c
	7461.0		
Schema	7761.2	302.16	< 1e – 15****
Gender ^d	7470.7	13.64	0.001**
Session-Grade	7463.0	3.96	0.046*
Student	8003.2	544.16	< 1e – 15****

^a Akaike Information Criterion for the model evaluated as $-2 \cdot (\log\text{Lik} - \text{npar})$. Smaller is better.

^b LRT statistic; twice the difference in log-likelihood, which is asymptotically chi-square distributed.

^c p value = $Pr(> \chi^2)$

^d Gender in (1 + Gender | School)

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, **** $p < 0.0001$

Grey shadings indicate statistically significant fixed effects

a model with gender as a fixed effect (M6). The LRT results in Table 12.8 show that introducing gender as a fixed effect (M6) does not significantly improve the model compared to the reduced model (M5) ($p = 0.4270$). However, the inclusion of gender as a random slope within schools (M4) significantly enhances the model fit ($p = 1e-3^{***}$), emphasising that the impact of gender on CAT scores varies across different school environments. This underscores the importance of considering the interaction between gender and the school context when assessing its effect on educational outcomes.

Table 12.8 – LRT to evaluate the global effect of Gender on the full dataset.

Comparison between the reduced model (M5) without the gender predictor, the model (M6) that considers gender a fixed effect, and the initial model (M4) that considers gender as a random slope within schools.

Model	AIC ^a	χ^2	p ^b
M5	7472.2		
M6	7473.6	0.631	0.4270
M4	7462.5	13.063	0.0003***

^a Akaike Information Criterion for the model evaluated as $-2 \cdot (\log\text{Lik} - \text{npar})$. Smaller is better.

^b p value = $Pr(> \chi^2)$

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, **** $p < 0.0001$

Grey shadings indicate statistically significant fixed effects

Chapter 13

Competencies assessment with IAS

The content of this chapter has been adapted from the following articles with permission of all co-authors and publishers:

- Mangili, F., **Adorni, G.**, Piatti, A., Bonesana, C., and Antonucci, A. (2022). Modelling Assessment Rubrics through Bayesian Networks: a Pragmatic Approach. In *2022 International Conference on Software, Telecommunications and Computer Networks* (SoftCOM) [181].
- **Adorni, G.**, Mangili, F., Piatti, A., Bonesana, C., and Antonucci, A. (2023a). Rubric-based Learner Modelling via Noisy Gates Bayesian Networks for Computational Thinking Skills Assessment. *Journal of Communications Software and Systems* [9].

As an author of these publications, my contribution involved:

Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft & review & editing, Visualisation.

13.1 Summary

This chapter focuses on using our IAS to provide a more holistic evaluation of students' competencies, contributing primarily to RQ3. Specifically, we apply the model to the data from the CAT unplugged, while for the virtual CAT, we integrate it into our tool for real-time assessment. We present the results and compare them with the task-specific assessment provided by the standard CAT score. Additionally, we compare the findings between the unplugged and virtual CAT assessments.

13.2 Methodology

The standard assessment, reflected by the CAT score (see Tables 6.1 and 7.3), is task-specific and evaluates the pupil's competence in completing individual tasks within a specific schema. While it provides insight into task performance, it does not offer a holistic view of the pupil's overall skills. In contrast, the assessment provided by the IAS delivers a more comprehensive evaluation of the pupil's performance. The BN computes probabilistic outputs, or posterior probabilities, for each target skill, as well as any relevant supplementary skills. These probabilities represent the likelihood that the pupil has mastered each skill based on their responses throughout the tasks. The BN-based CAT score is derived from these posterior probabilities and provides a global evaluation of the pupil's skill level. It is calculated by summing the marginal posterior probabilities of the target skill nodes, offering an estimate of how many competence levels the pupil has mastered. This summary score reflects a broader, more nuanced assessment of the pupil's abilities, aggregating the inferences drawn from all 12 tasks and accounting for both task performance and the model's estimation of the pupil's skill mastery.

To compare the standard assessment with the one obtained from the IAS, we analyse the correlation between the *BN-based CAT score* obtained with the IAS and the *average CAT score*, a baseline measure calculated as the mean of the individual task-specific CAT scores, for both the unplugged and virtual CAT versions.

Additionally, we present the inference times for the various models used in both the unplugged and virtual CAT versions. This information illustrates the computational efficiency of the BN-based evaluation process when comparing the different models.

To further investigate the models' effectiveness, we focus on a set of competence profiles selected from a group of students with interesting or notable performance patterns, comparing their average CAT score with the BN-based CAT scores derived from each model. This comparison highlights how the BN-based evaluation provides a more holistic and detailed understanding of pupil competencies. Furthermore, we also present the posterior probabilities for each skill based on the student responses. These probabilities represent the models' estimation of the pupil's proficiency, offering an overall estimate of skill mastery.

13.3 Evaluation of the unplugged CAT data

To evaluate model reliability and consistency with expert-based evaluation, we processed the responses of the 109 pupils from the unplugged CAT experimental study, calculating the posterior probabilities for the 9 target skills and, depending on the

model, for the 10 supplementary skills. Figure 13.1 illustrates the correlation between the average CAT score and the BN-based CAT scores for the following models: the baseline model (Model B), the baseline model with constraints (Model BC), the one which also includes the supplementary skills (Model BCS) and finally the enhanced model including both constraints and supplementary skills (Model ECS). The BN-based CAT score, originally in the $[0, 9]$ range, has been rescaled in the $[0, 4]$ range to align with the CAT score, defined on this scale, for easier and more direct comparison. In all cases, the Pearson correlation coefficient (ρ) is very high, indicating a strong consistency between the BN-based and the expert-based assessment.

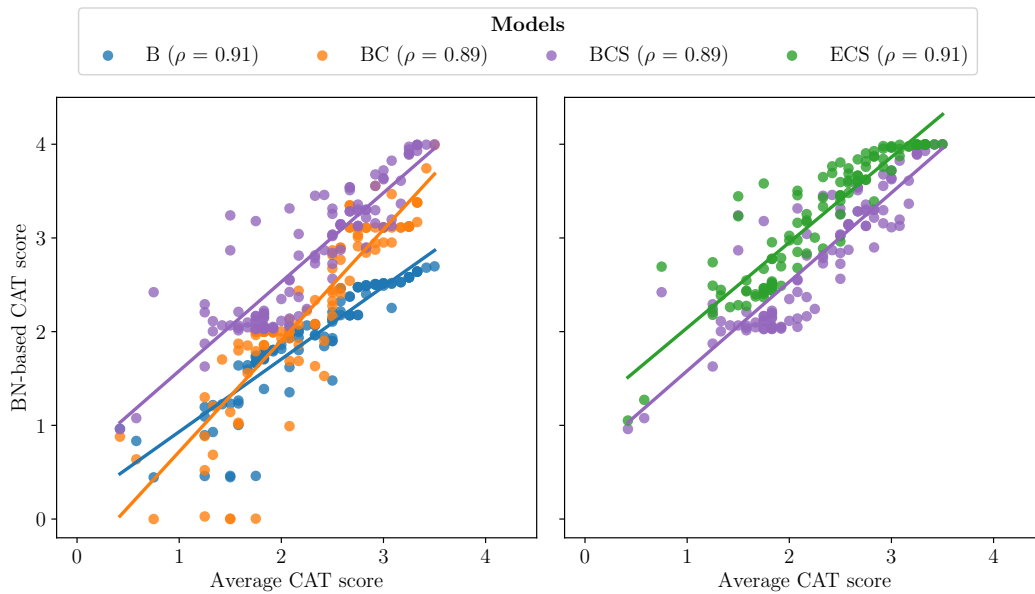


Figure 13.1 – Comparison of BN-based and average CAT scores (unplugged CAT).

Scatterplots showing Pearson correlation coefficients ρ between BN-based CAT scores of the four models and the average CAT score. On the left, the three baseline models are compared, and on the right, the baseline model with constraints and supplementary skills is compared to the enhanced model.

The inference times in Table 13.1 reveal significant variation across the four models. The models that assess only standard skills (Model B and Model BC) have relatively short inference times, with the total time for these models remaining under 30 seconds. In contrast, the models that incorporate supplementary skills (Model BCS and Model ECS) experience considerably longer inference times, exceeding 5 minutes for the total assessment. This increase in time is expected, as the addition of supplementary skills leads to a more complex model with a larger parameter space. Despite the differences in total inference times, it is important to highlight that the average inference time

Table 13.1 – Comparison of inference times across models (unplugged CAT).

For each model are reported the total inference time, the inference time per student (calculated by dividing the total inference time by the number of students), and the inference time per task (calculated by dividing the inference time per student by the number of tasks).

Model	Total inference time	Inference time per student	Inference time per task
B	29.615s	0.272s	0.023s
BC	28.940s	0.266s	0.022s
BCS	316.555s	2.904s	0.242s
ECS	306.517s	2.812s	0.234s

per student remains relatively short across all models, consistently under 3 seconds. These results suggest that all models are computationally efficient enough for real-time application, providing timely assessments for individual pupils without compromising the quality of evaluation.

When examining the posterior probabilities for individual students, we can gain deeper insights into their competence profiles, showcasing the interpretability of the model. By comparing the competence profiles produced by the four models considered, we can highlight significant differences in their assessments. Table 13.2 reports the answers provided by four representative pupils, allowing for a more detailed understanding of their performance. Table 13.3 compares the BN-based CAT scores to the traditional CAT scores, while Tables 13.4 and 13.5 display the corresponding posterior probabilities inferred by the models for the target skills and the supplementary skills, respectively.

Pupil 21 is a high-performing student, in terms of CAT and BN-based scores, who consistently used complex interactions for all CAT schemas and primarily used 2D algorithms.

Pupils 33 and 81 cannot be considered high-performing since they failed to complete some of the CAT schemas. Pupil 33 solved only the first seven schemas, always using 1D algorithms and almost always relying on the VS artefact. In comparison, student 81 was successful in the first nine schemas, where he applied different algorithms and artefacts, but mostly the 1D-V. For both students 33 and 81, the BN-based CAT scores predicted by the four models vary significantly, indicating that the models may be producing different predictions of their abilities. The difference between the original and BN-based CAT scores is inconsistent across the models. For both students, the largest difference between the original and BN-based CAT scores is observed in Model B, which predicts a much higher score for both students. On the other hand, Model BC predicts a meagre BN-based CAT score close to 0 for both students, indicating that

Table 13.2 – Tasks answers for a representative set of pupils (unplugged CAT).

For each task (T1-T12) are reported the target and supplementary skills applied. Supplementary skills correspond to: S_1 - paint dot, S_2 - fill empty dots, S_3 - paint monochromatic rows or columns, S_4 - paint monochromatic squares, S_5 - paint monochromatic diagonals, S_6 - paint monochromatic L-shaped patterns, S_7 - paint monochromatic zig-zags, S_8 - paint polychromatic rows or columns, S_9 - paint polychromatic diagonals or zig-zags, and S_{10} - repetition of a pattern.

Pupil	T1	T2	T3	T4	T5	T6
21	1D-V S_2	1D-V $S_2; S_6$	2D-V $S_3; S_{10}$	1D-V S_3	1D-V $S_3; S_4$	1D-V S_6
33	1D-V S_2	1D-VS $S_2; S_6$	1D-VS S_3	1D-VSF S_3	1D-VS S_3	1D-VS $S_1; S_3$
81	1D-V S_2	1D-V $S_2; S_6$	1D-V S_3	1D-VS S_3	1D-V $S_3; S_4$	1D-V S_6
92	1D-V S_2	1D-V $S_2; S_6$	1D-V S_3	1D-V S_3	1D-V $S_3; S_4$	1D-V S_6
Pupil	T7	T8	T9	T10	T11	T12
21	2D-V $S_8; S_{10}$	2D-V $S_1; S_5; S_{10}$	2D-V $S_1; S_{10}$	1D-V $S_1; S_4$	1D-V S_1	1D-V $S_1; S_5$
33	1D-VS S_5	fail	fail	fail	fail	fail
81	2D-VSF $S_1; S_5; S_{10}$	0D-VS S_1	2D-V $S_1; S_{10}$	fail	fail	fail
92	0D-V S_1	0D-V S_1	0D-VSF S_1	1D-VS $S_4; S_5$	2D-V $S_1; S_{10}$	0D-V S_1

Table 13.3 – Comparison of the average CAT score and the BN-based CAT scores across models for a representative subset of pupils (unplugged CAT).

Pupil	Average CAT score	BN-based CAT score
21	3.30	2.23 (Model B)
		1.65 (Model BC)
		1.98 (Model BCS)
		2.00 (Model ECS)
33	0.75	2.00 (Model B)
		0.00 (Model BC)
		1.33 (Model BCS)
		1.47 (Model ECS)
81	1.75	2.90 (Model B)
		0.07 (Model BC)
		1.62 (Model BCS)
		1.82 (Model ECS)
92	2.50	1.77 (Model B)
		1.42 (Model BC)
		1.59 (Model BCS)
		1.79 (Model ECS)

this model may not be the most accurate for these particular students. This suggests that other models may be better suited for predicting their performance on the CAT.

Pupil 92's performance was strong, as he successfully completed all 12 tasks using different skill levels. He solved the first six schemas with the 1D-V skill, reduced the algorithm's complexity in the following ones, changed artefact for some of the more complex tasks, and applied the highest level skill, 2D-V, in a tricky schema. Regarding the BN-based scores, all four models predicted a lower BN-based CAT score for student 92 than the original CAT score, although the differences were not as large as those observed for students 33 and 81. This suggests that student 92 is a relatively strong performer overall, but there is potential for improvement in his performance.

For all students, the baseline model assigns posterior probabilities equal or very close to one to the most used skill levels but fails to recognise that they also possess lower level skills, to which rather small probabilities, eventually equal to zero, as for the worst performing students 33 and 81, are assigned. On the one hand, when the ordering between skills is explicitly imposed, this problem is solved: the probabilities of lower skills increase, and those of higher skills decrease. This may lead to an excessive penalisation of higher skills, as in the case of pupils 33 and 81, where, as a consequence of the repeated failures in applying even the lowest competence level, Model

Table 13.4 – Posterior probabilities for target skills across models for a representative subset of pupils (unplugged CAT).

Pupil	Model	X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{31}	X_{32}	X_{33}
21	B	0.50	0.51	0.67	0.51	0.57	0.96	0.59	0.83	0.80
	BC	1.00	1.00	1.00	1.00	1.00	1.00	0.69	0.38	0.07
	BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.95	0.92
	ECS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
33	B	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
	BC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	BCS	1.00	1.00	0.52	1.00	1.00	0.05	0.59	0.30	0.00
	ECS	1.00	1.00	0.69	1.00	1.00	0.39	0.63	0.33	0.03
81	B	0.03	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	BC	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.91	0.21	0.03
	ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.95	0.67	0.44
92	B	0.55	0.40	0.41	0.33	0.13	1.00	0.46	0.05	0.00
	BC	1.00	0.99	0.99	0.76	0.70	0.68	0.13	0.00	0.00
	BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.59	0.19	0.01
	ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.79	0.58	0.41

Table 13.5 – Posterior probabilities for supplementary skills across models for a representative subset of pupils (unplugged CAT).

Pupil	Model	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
21	BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.40	1.00	0.26	1.00
	ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.52	1.00	0.38	1.00
33	BCS	1.00	1.00	1.00	0.42	1.00	1.00	0.38	0.15	0.16	0.13
	ECS	1.00	1.00	1.00	0.43	1.00	1.00	0.42	0.21	0.22	0.19
81	BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.36	0.34	0.31	1.00
	ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.39	0.41	0.35	1.00
92	BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.34	0.31	0.31	1.00
	ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.36	0.35	0.35	1.00

BC decides for the total absence of the competencies under examination, returning a posterior probability of zero, even for the skills successfully used by the students in several schemas. These inferences look too severe for these situations, where an expert would rather attribute the errors to the specific difficulties of the failed tasks rather than the total lack of algorithmic skills. On the other hand, when the supplementary skills are included in the assessment (Model BCS and Model ECS), this issue is solved, and the result of the posterior inference is consistent with the hierarchy of competencies defined by the rubric and the observations collected. In this case, the model understands that the failure follows from a lack of the supplementary skills necessary to solve specific schemas with more complex algorithms and not from a lack of target skills.

For instance, according to Model BCS and Model ECS, pupil 21 is likely to miss monochromatic zigzags (S_7) and polychromatic diagonals and zigzags (S_9), justifying the failure in applying the possessed 2D competence in schemas related to these supplementary skills. Finally, employing more elaborate models, such as Model ECS, may, in some cases, reward the ability to apply high-level skills in more complex tasks, i.e., those assigned with higher inhibition probabilities, such as for pupil 92 who managed to solve schema T11, a difficult one according to the parameters' elicitation in Figure 8.4, using a 2D-V skill and thus 2D algorithms are given a much higher probability by Model ECS than by Model BCS.

13.4 Real-time evaluation of the virtual CAT data

For the virtual CAT, we considered only the enhanced model, including both constraints and supplementary skills (Model ECS). To assess its reliability and consistency with expert-based evaluation, we processed the responses of the 129 pupils from the virtual CAT experimental study, calculating the posterior probabilities for the 12 target skills and the 14 supplementary skills. Figure 13.2 illustrates the correlation between the average CAT score and the BN-based CAT scores for the enhanced model in both the unplugged and virtual settings. In this case, the BN-based CAT score, originally ranging from $[0, 12]$, was rescaled to the $[0, 5]$ range to match the CAT score. For the virtual CAT, the Pearson correlation coefficient (ρ) is slightly lower at 0.85 but still demonstrates a strong alignment with the expert-based assessment. This suggests that Model ECS is reliable and consistent with expert evaluation in both contexts. A lower correlation does not necessarily imply a poorer model but rather indicates that the model remains consistent, albeit with some minor differences in the alignment between the two evaluations.

The comparison of inference times in Table 13.6 reveals significant variation across

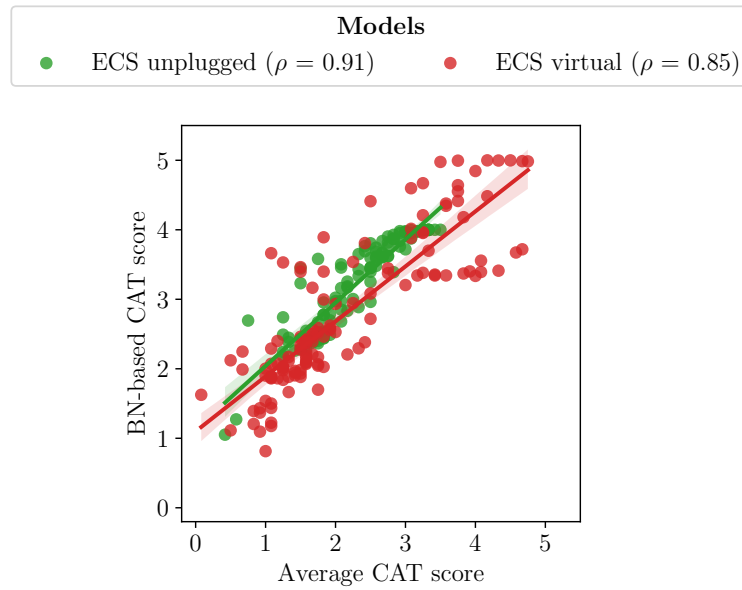


Figure 13.2 – Comparison of BN-based and average CAT scores (unplugged and virtual CAT).

Scatterplot showing Pearson correlation coefficients ρ between BN-based CAT scores of the enhanced model, for both unplugged and virtual CAT, and the average CAT score.

the two domains. For the unplugged CAT, the computational cost of the inference is relatively low, whereas, for the virtual CAT, the total inference time increases by a factor of approximately 10 (from 5 minutes to 53 minutes), with an inference time per student of 25 seconds and an inference time per task of 2 seconds. The higher inference time for the virtual CAT is primarily due to the increased complexity of the model, which includes a larger set of target skills (12 in total) and additional supplementary skills (14). This increase in the number of variables requires more computational re-

Table 13.6 – Comparison of inference times across models (unplugged vs virtual CAT). For the two enhanced models are reported the total inference time, the inference time per student (calculated by dividing the total inference time by the number of students), and the inference time per task (calculated by dividing the inference time per student by the number of tasks).

Model	Total inference time	Inference time per student	Inference time per task
ECS unplugged	306.517s	2.812s	0.234s
ECS virtual	3229.390s	25.034s	2.086s

sources and time to calculate the posterior probabilities for each student's performance. Although the inference times for the virtual CAT model appear significantly higher, it's important to note that the IAS does not perform post-hoc inference on the data as in the unplugged CAT. Instead, being integrated within the virtual CAT app, inferences are made in real time as students complete each task. This means they do not have to wait for the entire estimated time per student for the assessment to be processed, only for the estimated inference time per task.

We continue by analysing the model's results on a group of students to gain insights into their competence profiles. Table 13.8 reports the answers provided by four representative pupils, Table 13.7 compares their average CAT scores to the BN-based CAT scores, and Tables 13.9 and 13.10 display the corresponding posterior probabilities inferred by the models for the target skills and the supplementary skills, respectively.

Table 13.7 – Comparison of the average CAT score and the BN-based CAT score for the enhanced model for a representative subset of pupils (virtual CAT).

Pupil	Average CAT score	BN-based CAT score
5	1.50	2.34
89	0.08	1.63
39	2.42	3.81
65	4.50	5.00

Pupil 5 can be considered an average performer, with a CAT score of 1.5/5 assigned by the expert and a slightly higher score according to the IAS. This score suggests some level of competence, though Pupil 5 primarily relied on simpler 1D algorithms. The pupil only engaged with 2D twice and used a 0D algorithm once. Despite these limitations, he was able to successfully complete all tasks, relying on the gesture-based interface, often accompanied by visual feedback. This approach, while useful, could indicate a preference for more intuitive, immediate methods of interaction rather than a deeper engagement with different strategies or more abstract problem-solving techniques.

In contrast, pupil 89's scores are definitely lower, indicating a significant struggle with the tasks. This pupil failed multiple tasks and, when successful, consistently relied on basic 1D-GF strategies.

Pupil 39, on the other hand, achieved higher scores, indicating a high level of competence and application of skills. He effectively applied all interaction methods and produced algorithms of varying complexity, even creating some complex 2D algorithms for the most difficult tasks, demonstrating a diverse use of supplementary skills.

Table 13.8 – Tasks answers for a representative set of pupils (virtual CAT).

For each task (T1-T12) are reported the target and supplementary skills applied. Supplementary skills correspond to: S_1 - paint dot, S_2 - fill empty dots, S_3 - paint monochromatic custom patterns, S_4 - paint monochromatic rows or columns, S_5 - paint monochromatic squares, S_6 - paint monochromatic diagonals, S_7 - paint monochromatic L-shaped patterns, S_8 - paint monochromatic zig-zags, S_9 - paint polychromatic custom patterns, S_{10} - paint polychromatic rows or columns, S_{11} - paint polychromatic squares or L-shaped patterns, S_{12} - paint polychromatic diagonals or zig-zags, and S_{13} - repetition or copy of patterns, S_{14} - mirror patterns.

Pupil	T1	T2	T3	T4	T5	T6
5	1D-G S_2	1D-G S_3	2D-GF S_4	1D-G S_4	1D-G $S_3; S_4; S_5$	1D-GF $S_3; S_4; S_7$
89	1D-G S_2	1D-GF $S_2; S_3$	1D-GF $S_2; S_3$	1D-GF $S_2; S_3$	1D-GF $S_2; S_4; S_5$	fail
39	1D-PF $S_1; S_2$	0D-P S_1	1D-GF $S_1; S_4$	1D-PF S_4	1D-GF $S_4; S_5$	1D-G $S_3; S_7$
65	1D-P S_2	2D-P $S_2; S_4; S_{13}$	2D-P $S_2; S_4; S_{13}$	2D-P $S_2; S_4; S_{13}$	1D-P $S_4; S_5$	1D-P S_7
Pupil	T7	T8	T9	T10	T11	T12
5	0D-G S_1	2D-GF $S_1; S_6; S_8; S_{12}$	2D-GF $S_1; S_4; S_{10}$	1D-GF S_3	1D-GF $S_1; S_3; S_4$	1D-GF $S_1; S_6; S_8$
89	fail	fail	fail	1D-GF $S_3; S_4; S_5$	fail	fail
39	2D-PF S_{10}	2D-PF S_{10}	2D-PF S_{10}	1D-G $S_4; S_5; S_6; S_7$	2D-GF $S_1; S_4; S_7; S_{10}$	0D-GF S_1
65	2D-PF $S_{10}; S_{11}$	2D-P S_{10}	2D-P S_{10}	2D-PF $S_1; S_{11}; S_{12}; S_{13}$	2D-P $S_4; S_{10}; S_{14}$	2D-PF $S_8; S_{11}$

Finally, Pupil 65 achieved the highest score, indicating exceptional performance. This pupil demonstrated mastery across all tasks, applying a wide range of supplementary skills, mostly producing 2D algorithms using the most complex interaction methods.

Table 13.9 – Posterior probabilities for target skills in Model ECS for a representative subset of pupils (virtual CAT).

Pupil	X_{11}	X_{12}	X_{13}	X_{14}	X_{21}	X_{22}
5	1.00	1.00	0.53	0.19	1.00	1.00
89	1.00	0.71	0.45	0.22	1.00	0.28
39	1.00	1.00	0.99	0.85	1.00	1.00
65	1.00	1.00	1.00	1.00	1.00	1.00

Pupil	X_{23}	X_{24}	X_{31}	X_{32}	X_{33}	X_{34}
5	0.00	0.00	0.87	0.02	0.00	0.00
89	0.00	0.00	0.24	0.01	0.00	0.00
39	0.95	0.02	0.90	0.74	0.70	0.00
65	1.00	1.00	1.00	1.00	1.00	1.00

Based on posterior probabilities, given the observed data, we analyse the likelihood of each skill being present for each student. Pupil 6 demonstrates strong competence in 0D and 1D algorithmic tasks using the gesture interface. However, their ability to perform more complex 2D tasks is limited. They show minimal exploration of complex polychromatic patterns and lack proficiency in most 2D patterns, such as copying and mirroring. Successful application of 2D skills appears to be possible only when heavily relying on visual feedback.

Pupil 89 primarily uses the simplest interface, demonstrating strong performance in 0D algorithms, weak performance in 1D, and nearly no proficiency in 2D, with a clear preference for simple patterns and a sharp decline in competence with more complex patterns.

Pupil 39 demonstrates well-rounded competence, effectively using gesture-based interfaces to create algorithms of varying complexity, with high proficiency in nearly all 0D and 1D skills and good performance in 2D tasks. However, he is less skilled with the programming interface and can only create simple algorithms when supported by visual feedback. While he shows strong mastery of simple patterns, except for zigzag, his performance declines significantly for complex patterns and 2D-related skills.

Table 13.10 – Posterior probabilities for supplementary skills in Model ECS for a representative subset of pupils (virtual CAT).

Pupil	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇
5	1.00	1.00	1.00	1.00	1.00	1.00	1.00
89	1.00	1.00	1.00	1.00	1.00	0.32	0.44
39	1.00	1.00	1.00	1.00	1.00	1.00	1.00
65	1.00	1.00	0.51	1.00	1.00	0.49	1.00

Pupil	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄
5	1.00	0.32	1.00	0.32	1.00	0.29	0.29
89	0.40	0.27	0.28	0.27	0.28	0.25	0.25
39	0.42	0.35	1.00	0.35	0.31	0.28	0.28
65	1.00	0.38	1.00	1.00	0.39	1.00	1.00

Finally, pupil 65 demonstrates exceptional competence, with maximum probabilities across all skills, showcasing a comprehensive and versatile approach that includes advanced 2D patterns, with the exception of diagonal and custom patterns and interactions, highlighting his excellence in exploring complex and abstract strategies.

Part V

Discussion and Conclusion

Chapter 14

Summary and interpretation of findings

This chapter outlines the key findings of the study, with each subsection focusing on a research question, summarising results and comparing them to related work in the field.

14.1 Developing an age-based competence model for Computational Thinking

To answer the first research question, “How can a competence model for CT be defined to assess skills across different age groups and educational contexts?”, we developed two complementary frameworks.

To begin with, we adopted an alternative approach to classical CT models, which often emphasise individual cognitive abilities while overlooking critical factors such as the social and environmental influences and the developmental progression that shape cognitive processes during learning activities [125, 255, 258, 275, 299]. Based on the theory of situated cognition, we developed the CT-cube, a theoretical framework that extends existing CT models by addressing both the developmental and context-specific nature of CT. It integrates the cognitive processes essential for problem-solving with social and environmental factors, thus facilitating the design of CT activities that capture the multidimensional nature of CT and assessment of CT skills across different developmental stages and contexts. To validate its applicability, we applied this framework to design the CAT, an unplugged activity aimed at assessing AT in compulsory school pupils. The results indicate that the CT-cube effectively supports the assessment of CT skills in authentic classroom scenarios, highlighting its potential as a competence model for creating activities that reliably assess abilities across varied educational contexts and developmental stages.

Following this, we developed a second framework that builds upon the principles established in the CT-cube, aiming to address the lack of comprehensive competence models for CT from a different perspective. While many existing models focus on defining CT skills and competencies, they often fail to provide sufficient guidance on designing CTPs that effectively foster and assess these skills [43, 115, 171, 244, 262, 326]. Furthermore, many theoretical models are overly complex, incomplete, or overlap with one another [275, 299]. This second framework, FADE-CTP, focuses on identifying and analysing the components and characteristics of CTPs, recognising that the structure and context of these problems are crucial for CT skill development [125, 258]. To address gaps and overlaps in existing competence models, we created a structured catalogue of CT competencies that consolidates and organises the various models in the literature. This catalogue is linked to the components and characteristics of CTPs, facilitating the analysis of existing CTPs by identifying which competencies they can develop or assess based on their inherent characteristics. Additionally, it guides the design of new CTPs targeted at specific CT skills by outlining the necessary characteristics to activate their development.

In this way, both frameworks provide complementary approaches for defining and assessing CT competencies across different age groups and educational contexts, contributing to the development of more targeted and effective tools for fostering CT in educational settings.

14.2 Developing a large-scale assessment instrument for Algorithmic Thinking

To answer the second research question, “How can an activity and related instruments be developed to assess AT competencies on a large scale across different age groups and educational contexts, and what characteristics should they have to ensure their effectiveness and validity?”, building on the CT-cube and the FADE-CTP frameworks, we developed the CAT, an unplugged CT activity designed to assess algorithmic skills in compulsory school pupils. We tested the CAT in an experimental study and demonstrated its suitability for reliably measuring AT skills and providing insights into the progression of these skills across different developmental stages.

To enable large-scale assessment, we adapted the unplugged CAT into a digital format, creating the virtual CAT. This adaptation preserved the core problem-solving tasks and educational goals while allowing students to complete activities independently, reducing the need for administrator supervision and addressing scalability challenges. While in this form, the virtual CAT does not yet fully integrate IASs, its automated evaluation significantly improves efficiency and scalability compared to the unplugged

version, maintaining the integrity and quality of the assessment. This approach aligns with research highlighting the potential of technology-enhanced assessments to generate rich insights, support formative practices, and adapt to diverse educational settings [54, 284]. The virtual CAT was tested in two experimental studies, confirming its usability, accessibility, and versatility in catering to students across various developmental stages, backgrounds, and educational contexts. The platform encouraged active participation, with many students successfully completing their tasks. Moreover, the results demonstrated its efficiency in supporting large-scale assessments, meeting time and resource demands, and its potential for future integration with IASs to enhance scalability.

14.3 Developing an Intelligent Assessment System

To answer the third research question, “How can a probabilistic IAS be designed and integrated into the instrument for assessing AT skills across different age groups and educational contexts?”, we developed an IAS by translating a task-specific assessment rubric into a BN with noisy gates. BNs are widely recognised in recent literature as an effective method for modelling student knowledge. By leveraging the assessment rubric, our approach ensures a structured definition of relationships and parameters within the BN, enhancing its interpretability and applicability. Our implementation exploits the noisy gates mechanism to simplify parameter elicitation, making the system more efficient while preserving accuracy in assessing AT skills. Unlike conventional methods that assign a single score per student-task, our approach uses posterior probabilities to construct a comprehensive learner model that provides a more detailed understanding of students’ competence profiles, highlighting their proficiency across various skill levels [197, 249, 340, 341].

Specifically, we designed four BN-based models with increasing sophistication, starting with a simple baseline where all inhibitors share the same value. We then added constraints to model the ordering of competencies, incorporated supplementary competencies, and finally developed an enhanced model combining both features with expert-elicited parameters reflecting the intrinsic difficulty of tasks and competencies. The models were evaluated by comparing their assessments to expert evaluations, showing a high correlation and confirming the consistency of the BN-based assessments with expert judgments. Additionally, differences in posterior probabilities among the models highlighted the impact of improvements introduced during development, validating the iterative refinement process. Even the baseline model produced inferences closely aligned with expert assessments, suggesting that even a minimal parametrisation can serve as an effective starting point for further enhancements. Also, the

interpretability of the models was demonstrated through the analysis and comparison of competence profiles generated by the four models.

14.4 Examining Algorithmic Thinking competencies in Swiss educational settings

To answer the last research question, “What are the key AT competencies in the Swiss educational landscape, how do they develop across age groups, and what demographic or contextual factors are associated with variations in these competencies?”, we tested the unplugged CAT and virtual CAT activities in real-world classroom settings.

Our investigation of K-12 pupils’ algorithmic skills using the unplugged CAT revealed three key findings. (i) Algorithmic skills improve with age, particularly in autonomy and the ability to use more complex artefacts to describe algorithms. The most significant increase occurs between lower and upper primary school pupils, aligning with prior studies. For instance, Dietz et al. [83] demonstrate that older children become more successful and efficient at completing tasks, according to various measures, confirming the relationship between age and success rate. Similarly, Klahr and Robinson [160] observe improved problem-solving abilities and planning processes among older preschool children. (ii) Very young learners, such as those in preschool and lower primary school, are already capable of conceiving and describing complex algorithms, supporting the literature indicating the rapid development of AT skills in preschool-aged children [83, 208, 317, 318, 320]. This underscores the notion that complex problem-solving abilities can emerge earlier than previously thought [101, 154, 260]. (iii) There is no significant difference between genders with respect to the algorithmic skills of K-12 pupils, which contrasts with much of the existing literature. This finding could be specific to the context of our study, where no global gender differences were observed, possibly due to the interaction with other variables not measured. School-specific factors such as pedagogical methods, institutional culture, and student cohort dynamics may significantly influence performance variations across genders [236, 322]. The quality of instruction, classroom management, and local educational practices also play a key role [89, 323]. These results highlight the importance of considering local contexts, as academic achievements can vary across genders and regions [323].

Our investigation of K-12 pupils’ algorithmic skills using the virtual CAT revealed four key findings. (i) Algorithmic skills develop progressively with age, consistent with findings from the unplugged CAT. Younger students tend to use T&E strategies, especially with new tasks, but as they mature, they adopt more sophisticated problem-solving techniques and rely less on T&E. These results align with existing research on

the developmental progression of AT and problem-solving [75, 89, 164, 256, 313]. Like previous studies, our results confirm that younger students rely more on T&E when solutions are unclear, and their problem-solving methods become more advanced with age [53, 154, 304]. Our study further explores how T&E behaviour affects performance outcomes, noting that while excessive reliance on T&E may initially hinder performance, iterative attempts can promote learning, but relying solely on T&E without reflective thinking may limit deeper understanding and algorithmic competence [53, 275]. (ii) Our examination found no global gender effects on AT performance, which is consistent with the results from the unplugged CAT. However, we observed nuanced differences influenced by various factors. In the virtual CAT, simpler artefacts had less impact on algorithmic complexity for males than for females, while the unplugged CAT showed no significant gender differences. Age also moderated these differences, with males aged 10 to 13 outperforming females in unplugged settings but lagging in virtual ones. Furthermore, school performance data showed variability across institutions, with some showing higher performance for males and others for females. These findings suggest that gender's impact on AT performance is shaped by multiple factors, consistent with existing research [22, 164, 198, 229, 282]. The literature highlights the importance of early exposure to AT and effective teacher preparation to reduce gender gaps and promote equity [89] while emphasising the need for targeted interventions and supportive educational environments to address gender differences in early childhood [186]. These nuanced differences raise important questions about the factors contributing to variability and highlight the need to explore how some schools can better support all students, regardless of gender. (iii) The wide range of performances highlights the individual differences influenced by personal abilities, learning preferences, and external circumstances. This diversity underscores the need for equitable learning environments that accommodate various needs and learning styles, recognising that a one-size-fits-all approach may not be effective for all students. Addressing these differences is essential for ensuring that every student has the opportunity to succeed and develop their AT skills. This perspective aligns with research suggesting that tailored, personalised and adaptive educational approaches that address individual needs and characteristics can enhance learning experiences [80, 133, 191, 197, 278, 314]. By adapting educational practices to address diverse learning preferences and abilities, educators can create more inclusive and supportive environments that foster success for all students. (iv) Our analysis highlights the significant impact of different interaction modalities on the development of AT skills, revealing variations across age groups. Younger students predominantly use simpler artefacts, while older students shift to more complex artefacts, indicating a developmental progression toward more sophisticated problem-solving techniques. Interestingly, younger learners can also engage effectively with complex artefacts, suggesting

that exposure to such tools can foster advanced algorithmic skills at an earlier age than traditionally assumed. Interestingly, students using the virtual CAT generally demonstrated greater proficiency in advanced AT skills compared to those engaging with the unplugged CAT. The effectiveness can be attributed to the interactive and stimulating nature of digital environments, which provide a richer learning experience, aligning with theories emphasising the role of immersive learning environments in cognitive development [177, 179, 339].

Chapter 15

Practical implications

The CT-cube and FADE-CTP frameworks offer practical tools for assessing and developing CT skills in different educational contexts. The CT-cube helps design activities that account for both cognitive development and the context in which learning occurs, making it useful for evaluating CT skills in real classroom settings. The FADE-CTP framework focuses on identifying the key components of CTPs, helping to design tasks that target specific CT skills. It also allows for the evaluation of existing CTPs to determine which competencies they address. These frameworks provide guidance for educators and curriculum designers to create developmentally appropriate assessments and activities that foster CT skills across various age groups and learning environments.

The development of the unplugged CAT and its digital counterpart, the virtual CAT, offers practical benefits for educators and policymakers looking for scalable, efficient tools to assess AT skills across diverse student populations. The integration of a probabilistic IAS based on BNs enhances the precision and flexibility of assessments, supporting adaptive testing in educational contexts. The detailed evaluations provided by the system can guide instructional decisions, helping educators focus on competencies that need further development, especially in heterogeneous classrooms where students progress at different rates. Additionally, translating assessment rubrics into flexible mathematical models makes this approach accessible to educators with limited technical expertise, promoting the widespread adoption of IASs for real-time learner interaction. The effort required to create and refine the model is minimal, making this approach both scalable and easily adaptable to various educational environments, with the potential for further customisation with little additional effort.

The findings of this research have several practical implications for educators and policymakers in Switzerland and beyond. The developmental progression of AT skills observed in this study emphasises the importance of providing age-appropriate learning experiences that nurture these competencies. Introducing complex problem-solving

tasks earlier than traditionally thought could benefit younger students, who have demonstrated an ability to engage effectively with such challenges. The variability in performance, particularly regarding gender and contextual factors like school environments and artefacts used, highlights the necessity of tailored educational strategies. While global gender differences were not evident, the nuanced patterns observed suggest that addressing specific contextual variables is essential to promoting equity and inclusivity in AT education. Creating adaptive and inclusive learning environments that cater to diverse needs and learning styles is critical for supporting all students effectively. The use of digital tools, such as the virtual CAT, has shown promise in fostering cognitive development and algorithmic understanding, especially as students progress through different developmental stages. However, integrating these tools must be done thoughtfully, balancing their potential to enhance learning with the risks associated with excessive screen time. Research has shown that prolonged screen use can negatively affect cognitive, social-emotional, and physical development [203, 231, 285]. To mitigate these risks, educators should set reasonable limits on screen time and complement digital activities with other developmental approaches. By adopting a balanced, personalised, and inclusive approach, educational strategies can harness the benefits of interactive technologies while ensuring the overall well-being and success of learners.

Chapter 16

Limitations and Future works

This chapter discusses the limitations of this study and outlines potential directions for future research to address these challenges and build upon the findings.

16.1 Extending and validating the competence models

Despite the benefits and insights offered by our proposed competence models, further research is needed to validate and extend their applicability.

Specifically, the CT-cube framework, while promising, requires additional investigation to confirm its utility as a robust tool for the design and assessment of CT activities. Future studies should explore its application not only to algorithmic capabilities but also to problem setting tasks and more complex assessment scenarios. Moreover, expanding the research to other domains, such as educational robotics, and exploring more extensive regions, or even the entirety, of the CT-cube, for example, including formal artefactual environments, would be instrumental in assessing its full potential and versatility.

Although the FADE-CTP framework provides valuable guidance on the competencies that can be nurtured through specific CTPs, it does not yet address the levels of abstraction at which these competencies can be cultivated. Future work could focus on delineating whether these competencies emerge at foundational levels, such as recognising or understanding algorithmic concepts, or at more advanced stages, such as applying or synthesising them [36, 106]. Furthermore, broadening the framework to include more competencies, for example, those related to creativity, would greatly enhance its applicability. Such developments would support the design of more holistic CT activities and contribute to the advancement of educational strategies in this field.

16.2 Instrument potential for learning and teaching

The CAT holds promise not only as an assessment tool but also as an instrument for fostering the development of AT across various educational contexts. While traditionally employed for measuring students' algorithmic capabilities, CAT's structure allows it to be effectively repurposed for teaching and learning, thereby supporting both skill development and knowledge acquisition. As an instructional tool, CAT offers a versatile platform to engage students in active problem-solving activities that help them build foundational algorithmic concepts. Through iterative challenges and guided problem exploration, students can deepen their understanding of algorithmic structures and logical reasoning. The iterative nature of the tasks provides opportunities for students to experiment with different approaches, reflect on their solutions, and refine their strategies over time, promoting an active learning process. Moreover, integrating CAT into learning environments can facilitate personalised learning pathways. Its adaptable framework enables teachers to tailor tasks based on individual or group needs, helping to scaffold students' learning as they progress from basic concepts to more complex AT. The use of the tool in the classroom could further encourage collaborative learning, where students share their approaches, discuss their strategies, and learn from each other's solutions. In addition to supporting individual learners, CAT's use in classroom settings can enhance formative assessment practices. It allows instructors to monitor progress, identify areas of difficulty, and offer timely feedback to students, ensuring that each learner is appropriately challenged and supported. This ongoing interaction between assessment and learning can help bridge the gap between what students know and what they still need to learn, enhancing the overall effectiveness of the learning experience.

16.3 Integrating tutoring capabilities

A significant limitation of the current study is the absence of a tutoring mechanism, which was suggested during the expert evaluation by pedagogical experts but has not yet been implemented. Integrating tutoring capabilities into the BN-based IAS could transform it into a fully-fledged ITAS, providing real-time, adaptive support for students and greatly enhancing their learning experiences [80, 133, 191, 197, 278, 314]. The integration of tutoring functionality is essential for personalising the learning experience, offering tailored feedback and guidance to students in real time. The tutoring mechanism would be designed to assess a student's progress, identify areas of struggle, and provide contextual support to help students overcome difficulties. This adaptive tutoring would ensure that the support offered is neither too basic nor too advanced

but is instead aligned with the individual's current understanding and learning needs.

An important aspect of integrating tutoring capabilities lies in determining when and how much tutoring to provide. Establishing thresholds for identifying when a student is struggling or needs additional assistance to progress will be essential. This could be based on performance patterns, such as consistently incorrect answers, a lack of progress in completing tasks, or extended periods of inactivity. Recognising these signs would enable the system to provide timely support to help the student overcome challenges and continue their learning journey. The amount of tutoring offered needs to be calibrated to avoid overwhelming the student with excessive guidance. The system should aim to strike a balance, offering enough support to aid the learner's understanding without diminishing the opportunity for independent problem-solving.

In the context of the CAT, tutoring capabilities could guide students through the activity in various ways. For instance, the system could suggest adjustments to the type of interaction with the platform, such as recommending a shift from the gesture interface to the programming interface or vice versa. It could also activate visual feedback or suggest restarting a task if the student reaches a dead-end, helping them move forward in the learning process. Additionally, the system could suggest specific changes to the algorithm, such as recommending which blocks or commands to use, helping students refine their approaches and deepen their understanding of AT.

Alternatively, another promising avenue could be transforming this assessment into an adaptive test, where tasks are adjusted based on the student's performance. By measuring the information gained after each task, the system could identify areas of uncertainty and present the most relevant tasks to assess those areas instead of continuing to test already mastered skills. This approach would ensure that students are always challenged according to their current level, making the assessment more efficient and accurately measuring their algorithmic skills [20].

Currently, our system employs BNs with noisy gates for assessment purposes, where the probabilistic relationships between competencies are used to evaluate a student's performance and progress. To extend this framework for tutoring capabilities, the existing BN-based noisy gates would need to be adapted to provide real-time, personalised support. This could involve using the probabilistic model to identify when a student is struggling and dynamically adjust the intervention level, such as offering hints, feedback, or suggesting alternative problem-solving strategies. An alternative approach can rely on Structural Causal Models (SCMs) to explicitly model causal relationships between student actions, task difficulty, and learning outcomes. Unlike BNs, which model probabilistic dependencies, SCMs allow for a more direct understanding of how specific interventions, such as providing feedback or adjusting task difficulty, impact student performance. By modelling these causal effects, the tutoring system can predict the likely outcomes of different actions and select the most effective intervention based

on the student's current state. Finally, we developed a method for another project to find optimal solutions to the CAT problem [60]. This approach, which combines clustering, random search, and reinforcement learning techniques such as Proximal Policy Optimization (PPO), can also be leveraged to integrate tutoring functionalities into our instrument. The method could be incorporated into our developed app to suggest the best move for the student based on their progress so far. Since the system evaluates all possible moves and selects the one that maximises the number of coloured dots, it can be used to provide real-time guidance, recommending the most effective strategies and actions to help the student advance and solve the problem more efficiently.

16.4 Evaluating predictive power of the Intelligent Assessment System

In our study, to gauge the effectiveness of the IAS, we have primarily compared the model's outcomes with expert assessments. While this provides some validation of the system's utility in capturing relevant skills, a more robust evaluation of its predictive capabilities remains an important area for future development. One potential approach is to simulate the model's ability to predict answers to new questions based on prior responses. For instance, by randomly selecting 8 out of 12 schemas and conditioning on those responses, we could predict the student's answers to the remaining tasks. To achieve this, we would need to modify our current BN-based noisy gates model, shifting from skill-based inference to answer-based inference. After predicting the answers, we would reapply skill-based inference to generate the assessment based on these predicted answers. Finally, to assess the accuracy of this predictive approach, we could compute a loss function that measures the discrepancy between the predicted and actual answers for the remaining schemas. This evaluation would provide valuable insights into the model's ability to anticipate student performance and improve its overall predictive capabilities.

16.5 Instrument extension and validation

A key limitation of this study is that the instrument has not been fully validated to assess its effectiveness in capturing and measuring AT skills within real-world educational settings. Without proper validation, we cannot make definitive claims about its ability to assess AT skills reliably across diverse contexts. To address this limitation, future work should focus on validating the instrument by assessing its reliability in terms of face validity, content validity, and construct validity, ensuring it accurately

measures the relevant AT skills. Additionally, comparisons of the assessment results with those obtained from other established tools should be conducted to evaluate the instrument's relative effectiveness. Expert evaluations from professionals in the field of AT are also necessary to ensure the instrument accurately measures the skills it aims to assess. Additionally, pre-post intervention analyses can be conducted to examine the tool's effectiveness over time, assessing how well it captures changes in AT skills before and after an intervention. A large-scale study would also be crucial for evaluating the instrument's generalisability and reliability across various educational environments.

Overall, the results of this research should be interpreted in light of the limited sample size and the specific region of Switzerland where data was collected. Further investigations with larger samples from different school systems are needed to confirm the observed development of algorithmic skills. Additionally, the study did not explore socio-economic factors, such as parental income or education, which could influence performance. It also did not account for students' prior digital education, which might affect their ability to engage with the assessment and benefit from the instructional strategies tested. Future research should consider these factors to understand better how prior digital experience impacts AT development.

Finally, future research should explore how students' self-perception, interest, and motivation affect their performance in AT, as there is a strong link between high student engagement, a positive perception of the learning environment, and increased academic success [29, 34, 89, 118, 129, 163, 186, 212, 236, 273, 282, 289, 324]. Additionally, administering questionnaires to the teachers of these students could provide valuable insights into other contextual factors that may influence performance. Teachers' perspectives on students' learning habits, classroom dynamics, and individual challenges could shed light on external influences that might affect students' AT development. Moreover, it is important to consider that teachers may have biases, conscious or unconscious, which could impact how pupils learn and develop skills and, consequently, their assessment. Understanding these potential biases is crucial for interpreting the results accurately and ensuring that such factors do not influence the assessment tools and methods used.

16.6 Instrument integration

Integrating our assessment instrument into schools requires careful planning to ensure its effectiveness and sustainability. First, it is essential to collaborate with educators to ensure that the tool aligns with the curriculum and educational goals. We have already involved teachers in the development and testing phases, gathering valuable feedback on how the tool can be used effectively in real classroom settings. However, providing

professional development for teachers will be crucial, as they will need to understand how to interpret the results and use the tool to inform their teaching practices. Additionally, the integration should be seamless, causing minimal disruption to existing classroom routines. Technical support should also be readily available to ensure that both teachers and students can easily use the tool. To enhance usability, it may be necessary to make the tool more adaptable, allowing flexibility in its use and enabling teachers to customise it according to their students' needs.

Limitations related to access to technology still pose a challenge, especially for students without regular access to electronic devices or stable internet connectivity, as well as those with limited technological skills. In the studies conducted so far, we have already provided the necessary devices and infrastructure to ensure all participants can engage with the platform. However, integrating the platform into regular classroom settings could present difficulties, particularly in schools with limited or inconsistent access to technology. To address this, future studies could explore strategies for ensuring equitable access, such as collaborating with schools to provide devices or designing the platform to be more compatible with a variety of devices and internet conditions. Additionally, offering training to students with limited technological skills could help reduce disparities and facilitate more equitable participation in digital assessments.

Chapter 17

Conclusions

This thesis presented a comprehensive effort to advance the field of CT by addressing critical gaps in its assessment and analysis. The main objectives were to (i) develop an age-based competence model for CT, (ii) create a large-scale assessment instrument for AT, (iii) design an IAS, and (iv) examine AT competencies in Swiss educational settings, as measured using the developed tools. Each of these objectives has been systematically addressed through the work presented in this thesis.

The development of two distinct frameworks has provided structured approaches for defining and analysing CT competencies and designing related assessment activities. These frameworks offer a foundation for age-specific competence modelling and have been applied in the creation of assessment activities tailored to different developmental stages. Secondly, a large-scale assessment instrument for AT was developed in unplugged and virtual formats to ensure accessibility and adaptability in diverse educational contexts, facilitating widespread implementation in real-world settings. Thirdly, we designed an IAS powered by BN-based noisy gates to enhance traditional assessment methods based on experts' knowledge, introducing a probabilistic approach to assessing AT and providing detailed insights into students' reasoning processes and performance. Finally, the comprehensive analysis of data collected through this research has provided a detailed overview of AT competencies in Swiss educational settings, their development, and the factors that may influence learning outcomes.

By addressing these objectives, this work lays the foundation for further research and applications in CT and AT education. The tools and frameworks developed have the potential to be integrated into classroom practices, supporting educators in assessing and fostering CT skills. Moreover, it also opens pathways for future studies to validate and expand upon the developed tools, explore their applicability in different educational systems, and investigate additional factors influencing AT development.

Part VI

Appendices

Appendix A

Participant information sheets and parental consent forms

This appendix presents the documents related to the study's participant information and parental consent. The participant information sheets detail the purpose of the study, the procedures involved, and the participants' rights. First, the information sheet for the unplugged CAT study is presented, followed by the corresponding parental consent form. Then, the information sheet for the virtual CAT study and its parental consent form are included. The actual signed consent forms are kept on file separately for documentation purposes.

Information sheet for the unplugged CAT experimental study

Dear Parents,

On the day xxx, we will be present in Class xxx of School xxx in xxx to conduct an activity with the xxx to collect data for a research project funded by the Swiss National Science Foundation. The project is part of the National Research Programme on Digital Transformation (NRP 77).

Information about the program is available here: <https://www.nfp77.ch/en>.

Our project, carried out in collaboration with the Swiss Federal Institute of Technology in Lausanne (EPFL), the Dalle Molle Institute for Artificial Intelligence Research (IDSIA), and the University of Teacher Education St. Gallen, aims to develop methodologies for assessing computational thinking skills of children in compulsory schooling.

More details about the project are available at the following link: <https://www.nfp77.ch/en/ZfXBRNWkwXfpS1Co/project/are-our-children-developing-computer-complementary-skills>.

In particular, during the workday, I (the undersigned) and researcher Giorgia Adorni, a doctoral student at IDSIA and possibly other researchers active in the project, will be present from DFA, IDSIA, or EPFL. Students will engage in a simple activity: a student will observe a series of coloured dots arranged in the shape of a cross and describe verbally, through gestures, or via drawings the pattern to a peer or researcher who must reproduce it. The objective is to assess the ability of the xxx to recognise patterns in the dot arrangement and produce instructions that describe them effectively and efficiently.

An example of the pattern that will be presented to the xxx is as follows:



The data collected will consist of detailed instructions provided by the student for each pattern and any aids used during the activity. The data will be stored anonymously on secure servers at SUPSI with only an indication of the child's age (in years and months) and gender who produced them. The collected data will be analysed exclusively by researchers active in the project and will not be made available to third parties.

For any information regarding the project or the activities scheduled for the day xxx, you can contact me at phone number xxx or via email at xxx.

I would appreciate it if you could indicate your consent or refusal regarding your child's participation in the planned activity in the "Parental consent form" sheet and return the letter to your child to hand it over to the teacher.

Thank you very much for your cooperation and best regards,

xxx

Parental consent form

Data collection activity within the research project ``Assessing computational thinking skills with intelligent tutoring systems'' at Class xxx of School xxx in xxx.

Student's full name:

Please indicate your choice with a check mark.

I hereby declare that I have read the information regarding the project, and I communicate the following decision:

- ☐ I **agree** to allow my child to participate in the activity.
- ☐ I **prefer** that my child does not participate in the activity.

Place, Date, and Signature of parental authority:

Information sheet for the virtual CAT experimental study

Dear Parents,

On the day xxx, we will be present in Class xxx of School xxx in xxx to conduct an activity with the xxx to collect data for a research project funded by the Swiss National Science Foundation. The project is part of the National Research Programme on Digital Transformation (NRP 77).

Information about the program is available here: <https://www.nfp77.ch/en>.

Our project, carried out in collaboration with the Swiss Federal Institute of Technology in Lausanne (EPFL), the Dalle Molle Institute for Artificial Intelligence Research (IDSIA), and the University of Teacher Education St. Gallen, aims to develop methodologies for assessing computational thinking skills of children in compulsory schooling.

More details about the project are available at the following link: <https://www.nfp77.ch/en/ZfXBRNWkwXfpS1Co/project/are-our-children-developing-computer-complementary-skills>.

In particular, during the workday, researcher Giorgia Adorni, a doctoral student at the Faculty of Computer Science at the University of Italian Switzerland, and possibly other researchers active in the project will be present. Students will engage in a simple activity: with the help of an application on an iPad, they will individually observe a series of coloured dots arranged in the shape of a cross and describe it using gestures or through a specially designed block programming language. The objective is to assess the ability of the xxx to recognise patterns in the dot arrangement and produce instructions that describe them.

The application will record in real-time the instructions given by the student. The only personal data associated with the collected data will be the age (in years and months) and gender of the student, the school, and the class attended. Names and surnames will not be collected. The same activity was successfully conducted in 2021, but on paper instead of using iPads, and the results were summarised in a scientific publication in a prestigious international journal, available online at the following link: <https://www.sciencedirect.com/science/article/pii/S2451958821001147>.

To comply with Swiss data protection regulations and practices and ensure security, although the collected information is not particularly sensitive, the data will be automatically transferred from the iPads to a computer in the classroom via a local network (therefore without passing through the internet); from the computer, they will later be transferred, via a secure network, to secure servers based in Switzerland (<https://www.switch.ch/it/services/drive>). The complete data will be analysed exclusively by researchers active in the project.

In accordance with current Swiss practices in open science (<https://www.snf.ch/en/dah3uC2QX95tfPNd/topic/open-science>), the data may later be made available to researchers from other international academic institutions. However, the indication of the school and class attended will be removed, making it materially impossible to trace back to the source of the data.

For any information regarding the project or the activities scheduled for the day xxx, you can contact me at phone number xxx or via email at xxx.

I would appreciate it if you could indicate your consent or refusal regarding your child's participation in the planned activity in the "Parental consent form" sheet and return the letter to your child to hand it over to the teacher.

The teacher will inform during the activity which students have been granted permission to participate, without disclosing their names.

Thank you very much for your cooperation and best regards,

xxx

Parental consent form

Data collection activity within the research project ``Assessing computational thinking skills with intelligent tutoring systems'' at Class xxx of School xxx in xxx.

Student's full name:

Please indicate your choice with a check mark.

I hereby declare that I have read the information regarding the project, and I communicate the following decision:

- ☐ I **agree** to allow my child to participate in the activity.
- ☐ I **prefer** that my child does not participate in the activity.

Place, Date, and Signature of parental authority:

Appendix B

Linking CTP characteristics to CT competencies

This appendix provides a detailed analysis of our framework, FADE-CTP, represented in Table B.1, illustrating the rationale behind mapping CTP characteristics to CT competencies.

We have organised this analysis according to the main levels of the activity dimension: problem setting, algorithm, and assessment. For each dimension, we first describe the link between skills and the required features, then the link between the skills and the characteristics that act as catalysts.

The CTP features we considered are the functionalities allowed to the problem solver by the tools, the property of the system, such as resettability and observability, and finally, the trait of the task, including the elements required to be found, the type of cardinality of the elements given and to be found, the presence of constraints and the type of representation of the algorithm.

Table B.1 – Comprehensive overview of the relationship between different CTP characteristics and CT competencies.

The table shows the relationship between the characteristics of CTPs (columns) and CT competencies (rows). The CTP features considered include the tools' functionalities, the system's properties, and the task trait.

	Tool functionalities								System				Task											
	Variables	Operators	Sequences	Repetitions	Conditionals	Functions	Parallelism	Events	System resettable	System not resettable	System (partially) observable	System not observable	Initial or final state to be found	Algorithm to be found	One-to-one cardinality	Many-to-one cardinality	Explicit elements	Implicit elements	Unconstrained elements	Constrained elements	Algorithm manifest	Algorithm latent	Algorithm written	Algorithm not written
Data collection	✓							+	+	+	+	+			+	+		+		+	+	+	+	+
Pattern recognition	+		+	✓*	+	+	✓*		+	+	+	+			+	+	+	+	+	+	+	+	+	+
Decomposition	+	+	✓*		+	+	✓*	+	+	+					+	+	+	+	+	+	+	+	+	+
Abstraction	✓			+	+	+	✓			+		+				+	+	+	+	+	+	+	+	+
Data representation	✓		+	+	+	+				+		+				+		+		+	+	+	+	+
Variables	✓	+	+	+	+	+	+	+			+	+		✓ ^F	+	+	+	+	+	+	+		+	+
Operators	+	✓	+	+	+	+	+	+			+	+		✓ ^F	+		+	+	+	+	+		+	+
Sequences	+	+	✓	+		+					+	+		✓ ^F			+	+	+	+	+		+	+
Repetitions	+	+	+	✓		+					+	+		✓ ^F	+	+	+	+	+	+	+		+	+
Conditionals	+	+			✓			+			+	+		✓ ^F	+	+	+	+	+	+	+		+	+
Functions	+	+	+	+		✓					+	+		✓ ^F	+	+	+	+	+	+	+		+	+
Parallelism	+	+					✓				+	+		✓ ^F	+	+	+		+	+	+		+	+
Events	+	+			+			✓			+	+		✓ ^F			+	+	+	+	+		+	+
Algorithm debugging	+	+	+	+	+	+	+	+	✓	✗	+			✓				+			✓	✗	✓ ^F	✗ ^F
System state verification									✓	✗	+		✓					+		✓ ^{SP}	✗ ^{SP}	✓ ^F	✗ ^F	
Constraints validation	+	+	+	+	+	+	+	+	✓	✗	+								✗	✓				
Optimisation	+	+	+	+	+	+	+	+	✓	✗	+													
Generalisation	✓		+	+	+	✓		+	✓	✗	+				+		+		+					

✓ indicates that the characteristics is required for the development of the competence.

✓* indicates that at least one of several characteristics in a group is required for the development of the competence.

✗ indicates that the characteristic prevent the development of the competence.

+

Blank cells indicate that the characteristic is irrelevant for the development of the competence.

B.1 Problem setting competencies

B.1.1 Characteristics required for competencies development

Starting from the problem setting skills, to activate the “data collection” competence, the only requirement is that the tools available allow the use and recognition of variables. Without variables, there would be nothing to collect data on. The “pattern recognition” competence requires the presence of repetitions or functions since they allow the identification of repeating patterns in the data. The “decomposition” competence requires the presence of functions or sequences that can be used to break down a complex problem into smaller more manageable components. The “abstraction” competence demands the presence of variables to represent key concepts and functions to encapsulate and reuse specific behaviour within a single, self-contained unit, simplifying the original task and allowing the problem solver to reason about the problem at a higher level of abstraction. Finally, the “data representation” competence requires only variables to represent data.

B.1.2 Characteristics supporting competencies development

Generally, the attributes of the problem not required directly to activate the skills can influence them in some way. In the case of the characteristics of the tools, for example, variables also play a role in “pattern recognition” and “decomposition”, as they can be used to store patterns or parts of a complex problem. Then, operators can be useful for the “decomposition” of the problem into smaller parts. At the same time, sequences can contribute to the processes of “pattern recognition” and “abstraction”, helping the problem solver to identify patterns or regularities in the data, as well as the key concepts or essential elements of a problem, but also in “data representation” to organise and present data in a clear and meaningful way. Repetitions can influence the activation of the problem setting skills of “decomposition”, “abstraction”, and “data representation” because they can make the task more complex thus requiring the problem solver to use these practices. Similarly, conditionals can help to structure and simplify a problem, making it more manageable and easier to solve, enabling “pattern recognition”, “decomposition”, “abstraction” and “data representation”. Functions can influence the activation of “data representation” by helping the problem solver organise and structure data. Parallelism can influence the activation of the problem-solving skill of “decomposition” as it allows for breaking the problem into independent sub-tasks that can be executed simultaneously. Finally, events can trigger “data collection” at a specific point in time. For the sake of the characteristic of the system, resettability allows the problem solver to start over and try different approaches to solving the

problem, thus stimulating can problem setting skills such as “data collection”, “pattern recognition”, and “decomposition”, as they can test different strategies and collect data on their effectiveness. On the other hand, if the system is not resettable, the problem solver may have to rely more on “abstraction” and “data representation” skills to find a solution, as they cannot try different approaches and must work with the information they have available. In general, a resettable system allows more freedom for the problem solver, giving a chance to explore different solutions. In contrast, a non-resettable system may require more creativity to find a solution. If the system is observable, the problem solver would likely use skills related to “data collection”, as he can directly perceive and then gather information about the system’s state and properties. Additionally, he may use skills related to “pattern recognition”, such as identifying patterns or trends in the data collected. These skills can help the problem solver understand the system’s current state and make informed decisions about how to solve the task. Conversely, suppose the system is not observable. In that case, the problem solver may need to rely on abstract and hypothetical reasoning to devise a solution, activating “pattern recognition”, “decomposition”, and “abstraction” to understand the problem and identify possible solutions. Also “data collection” may be necessary to gather information about the system and its behaviour, even if that information is not directly observable. Additionally, the “data representation” skill can be used to organise and interpret the information they have collected to make sense of the problem and develop a solution. When there is a many-to-one cardinality in the system, it means that there is a large amount of data that needs to be processed, and multiple inputs or sources of information can be used to achieve a single goal or outcome. In this scenario, the “data collection” skill will likely be activated because the problem solver needs to gather a large amount of information to understand the problem and find a solution. Since there are multiple instances of a certain element or pattern, recognising the commonalities and differences among them would be essential to understand the overall system, leading to the use of more complex data collection and analysis strategies, thereby activating the “pattern recognition” competence. The “decomposition” and the “abstraction” skills will also likely be activated as the problem solver needs to break down the problem into smaller manageable parts and find the underlying principles and concepts in the problem to understand the overall system and find a solution. Finally, as there are multiple instances of a specific element, it would be essential to communicate them clearly and concisely, thus activating the “data representation” skill. By contrast, if there is a one-to-one cardinality in the system, the competencies of problem setting that are likely to be activated include “data collection”, “pattern recognition”, and “decomposition”. The skills “abstraction” and “data representation” are less likely to be activated since the direct correspondence between the system elements means there is less need to abstract or represent the information.

It can be assumed that with implicit elements, the “data collection”, “pattern recognition”, “decomposition” and “abstraction” competencies may be activated as the problem solver needs to infer information from the context or the environment, understand the underlying concepts or patterns in the task, decompose the problem into smaller sub-problems, and create abstract representations of the system. The same reasoning can be applied to constrained elements. Moreover, it is possible that the competence “data representation” may be activated as implicit or constrained elements may require the problem solver to think about how to represent the data in a way that accurately reflects the underlying information or constraints. Likewise, with explicit elements, the “pattern recognition”, “decomposition”, and “abstraction” competencies may also be activated, as the problem solver needs to understand and make sense of the given information, and the presence of unconstrained elements to be found may allow for more flexibility and creativity in problem-solving, potentially activating these skills, as problem solvers may need to find novel ways to organise or make connections among the elements. In this scenario, the “data collection” and “data representation” competencies may be more straightforward and not as crucial, especially when the elements are explicit and thus the information is already provided in a structured format. Regarding the representation of the algorithm, overall all problem setting competencies may be activated. Nevertheless, a manifest algorithm makes the problem solver’s task easier by providing a clear set of instructions and reducing the need for “pattern recognition” and “decomposition”. However, a not manifest algorithm can promote more “pattern recognition”, “decomposition”, and “abstraction” as the problem solver needs to infer the algorithm from the problem statement and available information and cannot represent it.

B.2 Algorithmic competencies

B.2.1 Characteristics required for competencies development

For the algorithm dimension, each competence to be activated requires that the corresponding characteristic of the tool is enabled. For example, to activate the “variable” skill, the tools used by the problem solver should include variables. Moreover, in a formal artefactual environment, the task requires that the algorithm is not given but has to be found. Otherwise, it is possible only to assess the problem solver ability to recognise these skills and apply them, but not create an algorithm from scratch.

B.2.2 Characteristics supporting competencies development

Again, some characteristics can also influence the activation of algorithmic competencies. Regarding the characteristics of the tools, for example, the presence of variables may influence the activation of all the other algorithmic skills, since they provide a fundamental building block for creating algorithms and can be used in conjunction with other algorithmic structures. Similarly, operators influence the activation of all algorithmic skills. The presence of sequences may influence the activation of “variables”, “operators”, “repetitions” and “functions”; repetitions may influence the activation of “variables”, “operators”, “sequences” and “functions”; the presence of conditionals may influence the activation of “variables”, “operators” and “events”; functions may influence the activation of “variables”, “operators”, “sequences” and “repetitions”; the presence of parallelism may influence the activation of “variables” and “operators”; while events may influence the activation of “variables”, “operators” and “conditionals”. The resetability or non-resetability of a system is not relevant for activating or not algorithmic competencies. The system’s observability, or the ability to observe the agent’s actions and the system’s state, allows tracking of how the algorithm is executing and makes it easier for the problem solver to identify these procedures used by the agent. Instead, a non-observable system may activate the skills of “variables”, “operators”, “sequences”, and “conditionals”, since the problem solver may need to rely more heavily on their ability to reason about the system and make inferences based on limited information. Regarding the ratio of elements given and to be found, from one side, a one-to-one cardinality may influence the activation of the algorithmic skill “variables” that can be used to define the direct correspondence between the elements in the system and their representations, but also of “operators” and “conditionals” proper to manipulate them and necessary to ensure the correct mapping. On the other side, a many-to-one cardinality can make it more challenging to understand the relationship between the given elements and those to be found, impacting the ability to understand the algorithm and its parts and enforcing the use of certain types of structures. For example, the problem solver can keep track and map multiple instances to a single object using “variables”. If the task at hand involves processing multiple pieces of data and producing a single result, a “repetition” can be used to iterate over the inputs. Similarly, “conditionals” can be used if the task requires selecting one output out of multiple possibilities based on certain conditions. In contrast “functions” can be used to modularise the code and make it more organised and maintainable. Finally, “parallelism” can be used to speed up the processing of multiple inputs by running multiple iterations simultaneously. Further, explicit elements provide clear and specific information about the task that must be solved, allowing the problem solver to use all the algorithmic structures to manipulate and work with that information to achieve the desired outcome. Besides,

the presence of implicit elements in the task makes it more difficult for the problem solver to understand and determine the necessary steps to solve the task, thus some algorithmic structures may need to be used to compensate for this shortcoming. For example, “variables” would be necessary to store and track the values of implicit elements, “operators”, “sequences”, “repetitions”, “conditionals”, and “functions” would help make decisions and perform actions based on the values of these variables. These algorithmic structures would allow the problem solver to explain the implicit elements effectively and develop more sophisticated and efficient solutions. Similarly, the space for possible solutions is limited when constrained elements are involved in the task and it may be necessary to use some algorithmic structures to ensure those constraints are met. For example, while solving a puzzle, the final state and algorithm have to be found, and they have constraints: the problem solver has to fit several pieces together to form a complete image, pieces must fit together to form a specific figure, and certain pieces can only be placed in certain orientations. To solve this task, the problem solver might use a combination of algorithmic structures such as “variables” to keep track of the current state of the puzzle and the position of the pieces, “operators” to manipulate the pieces and move them around, “sequences” to try different combinations of pieces, “repetitions” to keep trying different combinations until the puzzle is complete, and “conditionals” to check if the current combination of pieces meets the constraints. Additionally, “functions” could also be used to group sets of repeated actions. Finally, how the algorithm is represented can affect the activation of various algorithmic structures depending on the type of representation used. Considering different types of tools, each can be more suited to activating one skill rather than another. If the algorithm is represented in a mathematical notation, the use of “operators” may be more prominent. On the one hand, if the algorithm is represented in a visual block-based programming language, the use of “sequences”, “repetitions” and “conditionals” may be more intuitive and easier to activate. On the other hand, if the algorithm is represented in a text-based programming language, the use of “variables” and “functions” may be more natural to activate. Finally, robotic programming languages are usually designed for detecting and responding to “events”, such as sensor readings or other inputs. They often have built-in functionalities for concurrent execution of multiple instructions, allowing “parallelism”. Overall, the choice of representation can affect the ease and familiarity of activating different algorithmic structures and may also shape the problem solver’s understanding and ability to apply them effectively.

B.3 Assessment competencies

B.3.1 Characteristics required for competencies development

Finally, in the assessment category, all skills have in common the need for the system to be resettable for the skill to be activated. For example, in “algorithm debugging”, if the instruction cannot be reversed, it is impossible to revise and test the previous code versions. Thus, resettability is necessary to debug the algorithm in a controlled and repeatable environment. The same applies to correcting errors in the state and constraints and improving the solution’s performance or generalising it. In the specific case of “algorithm debugging”, this skill can be activated in all the artefactual environments if the algorithm has to be found and if it is manifest because it allows the user to understand and check the logic and the flow of the algorithm. This is essential to identify and fix any bugs or errors in the algorithm. While it becomes increasingly important to have a written algorithm as the difficulty level of the artifactual environment rises, it may still be possible to solve the problem without one. However, the absence of a written algorithm may make it more challenging to analyse or modify the solution in a formal setting, as the artefactual environment is more abstract and requires a more in-depth understanding. For this reason, we considered the skill required in this context. The “system state verification” competence can be activated in all three artefactual environments if at least one between the initial and final states must be found. In embodied environments, direct physical interactions with the system provide a way to observe its state without needing a manifest algorithm. However, in symbolic and formal environments, a manifest representation of the algorithm, written in the case of formal environments, is crucial to fully understand its logical flow, verify the system state, and perform formal reasoning about its correctness. This may involve analysing the symbolic representation to understand how it impacts the system state. To activate the “constraints validation” competence, it is blatant that the other necessary characteristic is having constraints on the states to be found. To enable “optimisation”, additional features are not required, while for “generalisation”, variables and functions are necessary to reuse and apply the task solution to different problems.

B.3.2 Characteristics supporting competencies development

Each tool functionality available to the problem solver can be a potential cause of error in the algorithm. For example, if the problem solver is unfamiliar with one of them or does not understand how to use it correctly, he may not use it at all or misuse it. This can lead to errors in the algorithm and potentially result in the problem not being solved correctly. This is why functionalities of the tools if available can activate “algo-

rithm debugging”. Also for “constraints validation”, all the characteristics of the tools are influential. Above all, variables, operators, conditional and functions may allow the problem solver to perform various calculations and comparisons to check if the values assigned to the variables meet the specified constraints. Further, it could be that the constraint imposed is precisely on the algorithm and prohibits using some of these structures. The functionality of the tools available to the problem solver can greatly impact the “optimisation” of the algorithm in several ways. Parallelism allows for multiple tasks or processes to be executed simultaneously, which can greatly reduce the overall time required to complete a task. Sequences and other structures, such as loops, can also help to improve efficiency by allowing for the automation of repetitive tasks and the ability to perform actions in a specific order. Additionally, using functions and subroutines can improve the readability and maintainability of the algorithm, making it easier to identify and fix any errors that may occur. However, having access to a wide range of functionalities can make it challenging for the problem solver to choose the appropriate one for a specific task, leading to a revision of the solution to increase efficiency and performance. The competence “generalisation” can also be influenced by other characteristics of the tools. The presence of sequences and repetitions in the toolset enables the problem solver to apply the same algorithm to different parts of a problem or task. Similarly, the inclusion of conditionals allows for the application of different algorithms depending on the specific conditions of the task. Furthermore, the presence of events in the toolset allows for creating algorithms that can respond to different triggers within the problem, leading to a greater generalisation of the solution and the ability to adapt to changes within the problem. In terms of observability, an observable system allows the problem solver to have a clear understanding of the system’s state and the output of the algorithm, which can aid in identifying and addressing errors and inefficiencies and performance issues, as well as recognising patterns or regularities that can be generalised to new or different situations. However, it is essential to note that while observability can aid in all assessment skills, it is not strictly necessary for their activation. For example, one could still perform “algorithm debugging” and “system state verification” on a non-observable system, though it may be more difficult. Similarly, “generalisation” can still occur without perfect observability, but it may be harder to identify patterns and regularities without direct access to the system state. If the system has a many-to-one cardinality, the competence of “generalisation” may be activated as it would be necessary to apply the same algorithm to different inputs or outputs. If the system contains implicit elements, the competencies “algorithm debugging” and “system state verification” may be activated as the problem solver may need to identify and troubleshoot any issues with the algorithm that are not immediately apparent or infer the current state of the system based on the implicit information provided. Also “generalisation” may be activated as the problem solver may need to

apply the algorithm to different situations based on the implicit information provided. Finally, suppose in the system there are elements to be found with constraints. In that case, the “generalisation” skill may be activated because it requires the problem solver to adapt the task to the specific constraints and can be intended as solving a new problem using the knowledge acquired in a previous situation and adapting it to a new one.

Appendix C

Main study with the unplugged CAT

This appendix provides detailed documentation related to the main study with the unplugged CAT assessment, in particular we included: (i) the protocol template that outlines the experimental setup and recording procedures for the unplugged CAT assessment and is intended to facilitate replication of the study and ensure transparency in the methodology employed [226]; (ii) the illustrations of all the algorithms conceived by students for each schema used in the assessment.

C.1 Protocol template for administering the activity

This section outlines the protocol template used to guide the administration of the unplugged CAT assessment. It provides a detailed framework for the experimental setup, ensuring consistency and reliability in data collection.

For each participant, a separate protocol should be filled in, capturing general information such as Session ID, School ID/Name, Pupil ID, Date, Administrator, Class/Grade, Pupil Age, and Pupil Sex.

For each schema presented to the participants, the type of interaction used and the algorithm dimensions produced should be recorded. Finally, additional notes or observations related to each schema can be included to provide further context and insights into the participant's approach.

Cross Array Task (CAT) experimental protocol

Session ID:

Date:

Administrator:

School ID/name:

Class/Grade:

Pupil ID:

Pupil age:

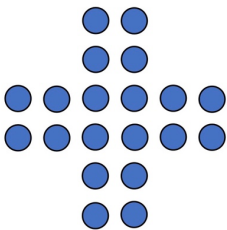
Pupil sex:

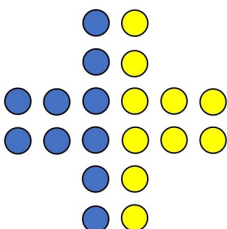
Type of interaction:

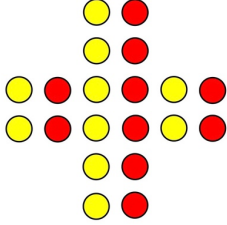
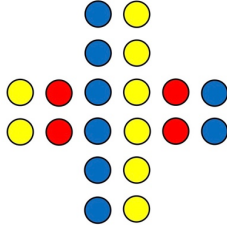
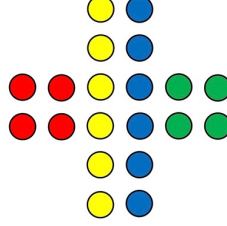
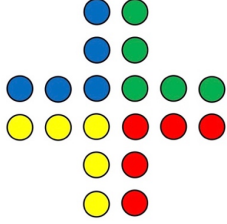
- V: using voice commands;
- VS: using voice commands and hand gestures on an empty cross array;
- VSF: voice commands and hand gestures on an empty cross array, hinging on visual feedback;

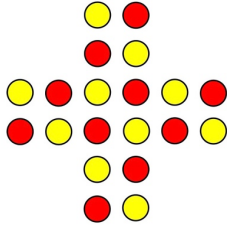
Algorithm dimension:

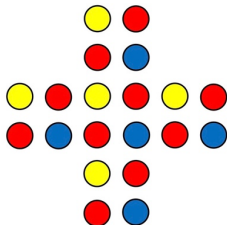
- 2D: polychromatic colouring of patterns of dots, repetitions of patterns and symmetries;
- 1D: monochromatic colouring of patterns of dots
- 0D: colouring of individual dots

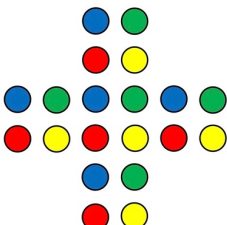
S1 	Type of interaction (artefact and autonomy)		
	V	VS	VSF
	Algorithm dimension		
	2D	1D	0D
	Observations and notes		

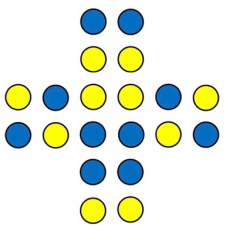
S2 	Type of interaction (artefact and autonomy)		
	V	VS	VSF
	Algorithm dimension		
	2D	1D	0D
	Observations and notes		

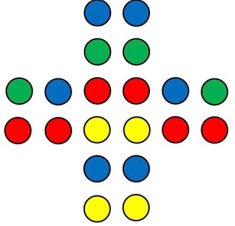
S3		Type of interaction (artefact and autonomy)		
		V	VS	VSF
		Algorithm dimension		
		2D	1D	0D
		Observations and notes		
S4		Type of interaction (artefact and autonomy)		
		V	VS	VSF
		Algorithm dimension		
		2D	1D	0D
		Observations and notes		
S5		Type of interaction (artefact and autonomy)		
		V	VS	VSF
		Algorithm dimension		
		2D	1D	0D
		Observations and notes		
S6		Type of interaction (artefact and autonomy)		
		V	VS	VSF
		Algorithm dimension		
		2D	1D	0D
		Observations and notes		

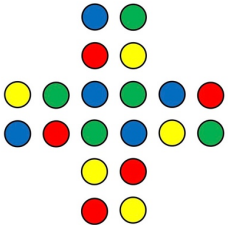
S7		Type of interaction (artefact and autonomy)		
		V	VS	VSF
		Algorithm dimension		
		2D	1D	0D
		Observations and notes		

S8		Type of interaction (artefact and autonomy)		
		V	VS	VSF
		Algorithm dimension		
		2D	1D	0D
		Observations and notes		

S9		Type of interaction (artefact and autonomy)		
		V	VS	VSF
		Algorithm dimension		
		2D	1D	0D
		Observations and notes		

S10		Type of interaction (artefact and autonomy)		
		V	VS	VSF
		Algorithm dimension		
		2D	1D	0D
		Observations and notes		

S11 	Type of interaction (artefact and autonomy)		
	V	VS	VSF
	Algorithm dimension		
	2D	1D	0D
Observations and notes			

S12 	Type of interaction (artefact and autonomy)		
	V	VS	VSF
	Algorithm dimension		
	2D	1D	0D
Observations and notes			

Supplementary observations:

C.2 Algorithmic solutions developed by students

This section focuses on the exploration of the algorithms conceived by the students for each of the cross array schemas used in the unplugged CAT assessment. For each schema, we present illustrations of the different algorithms designed by students. The first algorithm for each schema is always the zero-dimensional algorithm, which describes the array point-by-point. All other algorithms are assigned randomly and do not necessarily share any specific commonality. By examining the diversity of algorithms and their features, this section helps illustrate how students engage with algorithmic thinking in an unplugged context.

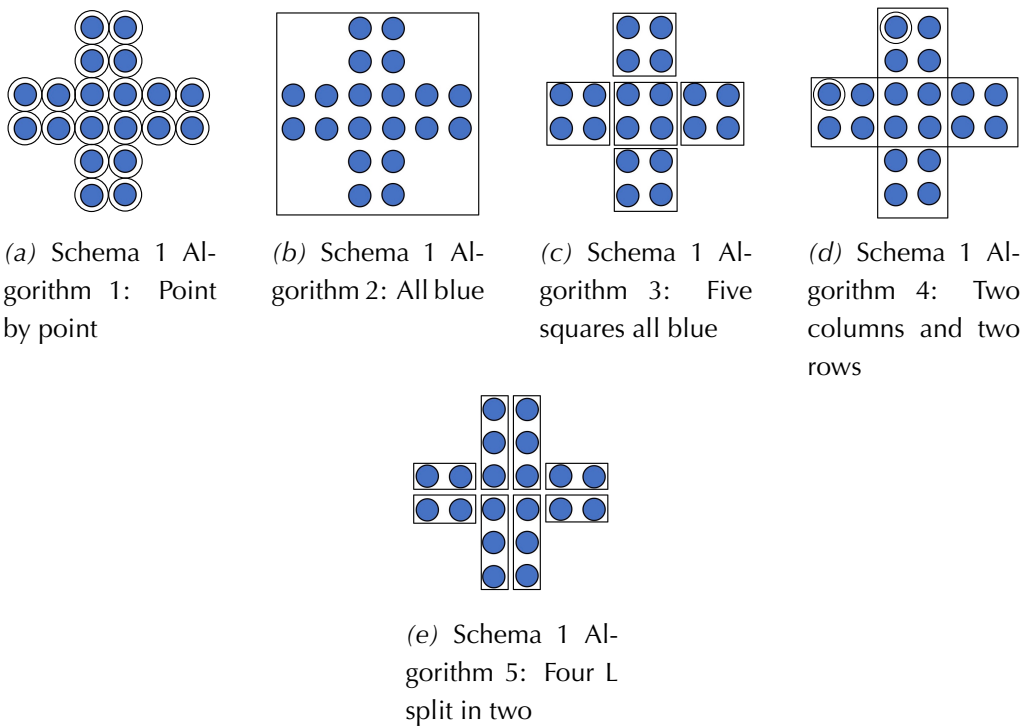
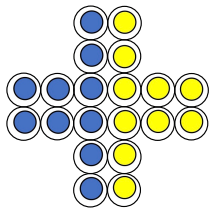
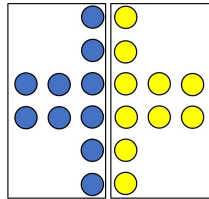


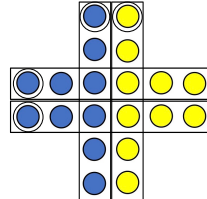
Figure C.1 – Algorithms observed for schema S1.



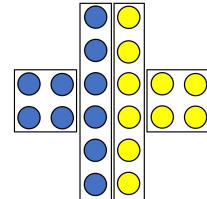
(a) Schema 2 Algorithm 1: Point by point



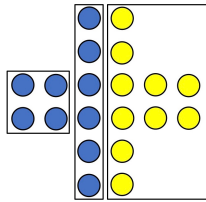
(b) Schema 2 Algorithm 2: Split vertically, blue on the left yellow on the right



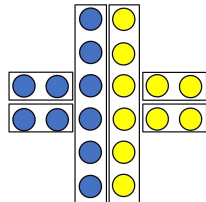
(c) Schema 2 Algorithm 3: Two columns and two rows



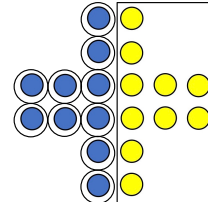
(d) Schema 2 Algorithm 4: Two columns and two squares on the sides



(e) Schema 2 Algorithm 5: Half of one colour, one square and one column with the other

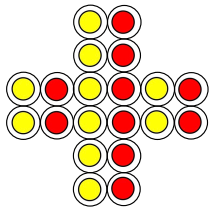


(f) Schema 2 Algorithm 6: Two columns and the sides row by row

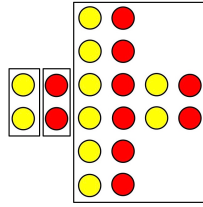


(g) Schema 2 Algorithm 7: Half point-by-point, the remaining of one colour

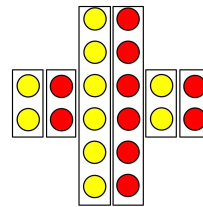
Figure C.2 – Algorithms observed for schema S2.



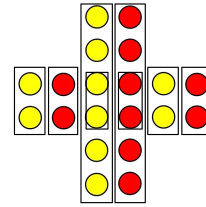
(a) Schema 3 Algorithm 1: Point by point



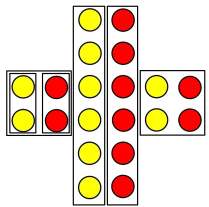
(b) Schema 3 Algorithm 2: Alternate columns



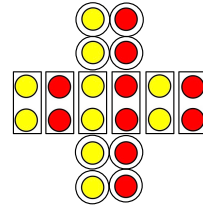
(c) Schema 3 Algorithm 3: Column by column



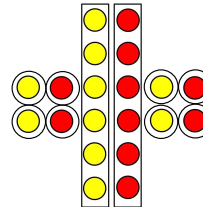
(d) Schema 3 Algorithm 4: Column by column (with redundancy)



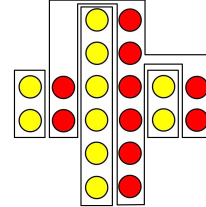
(e) Schema 3 Algorithm 5: Column by column symmetrical sides



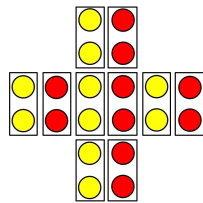
(f) Schema 3 Algorithm 6: Two rows column by column, point-by-point up and down



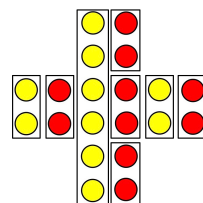
(g) Schema 3 Algorithm 7: Two columns, squares point-by-point



(h) Schema 3 Algorithm 8: One colour column by column, the remaining of the other colour



(i) Schema 3 Algorithm 9: Two rows column by column, the remaining column by column



(j) Schema 3 Algorithm 10: One column with pairs of two, the remaining column by column

Figure C.3 – Algorithms observed for schema S3.

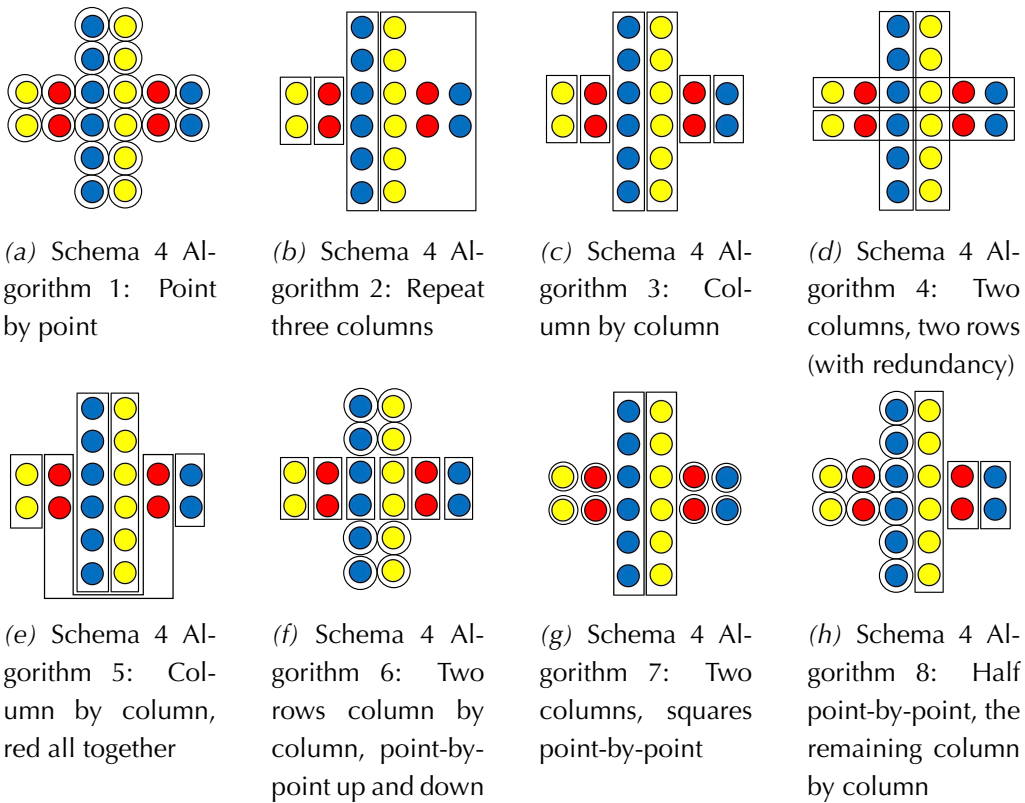


Figure C.4 – Algorithms observed for schema S4.

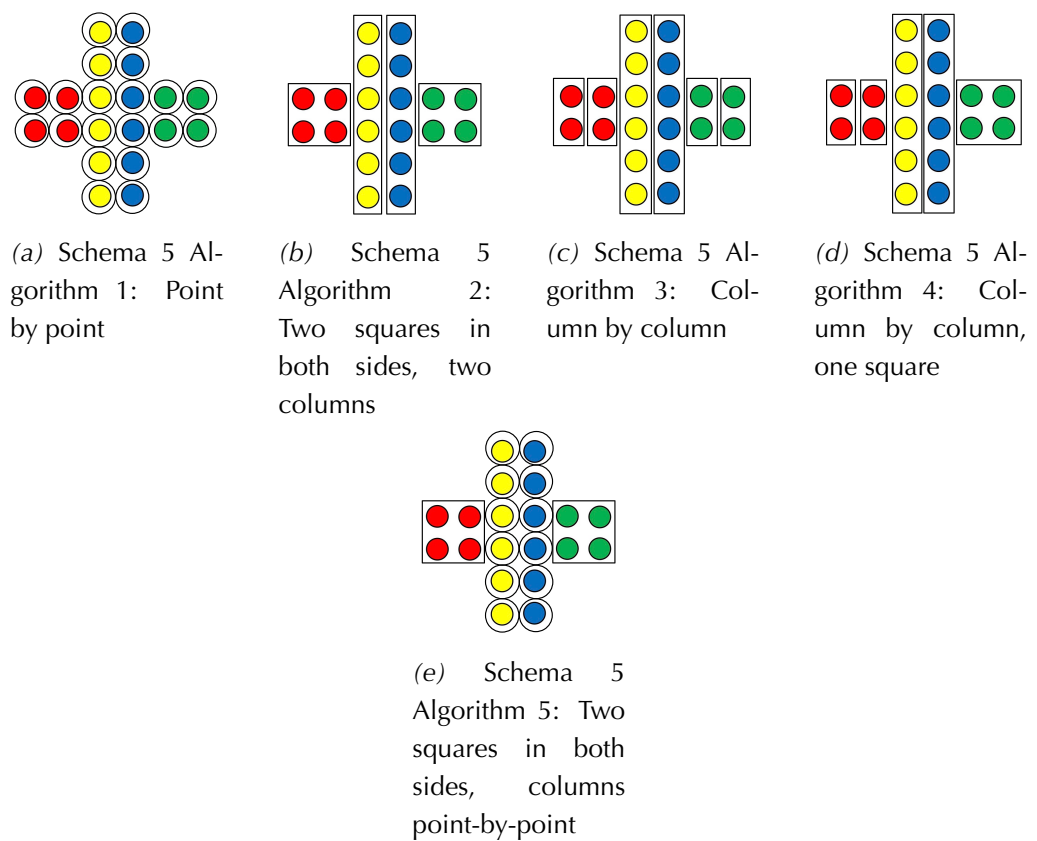


Figure C.5 – Algorithms observed for schema S5.

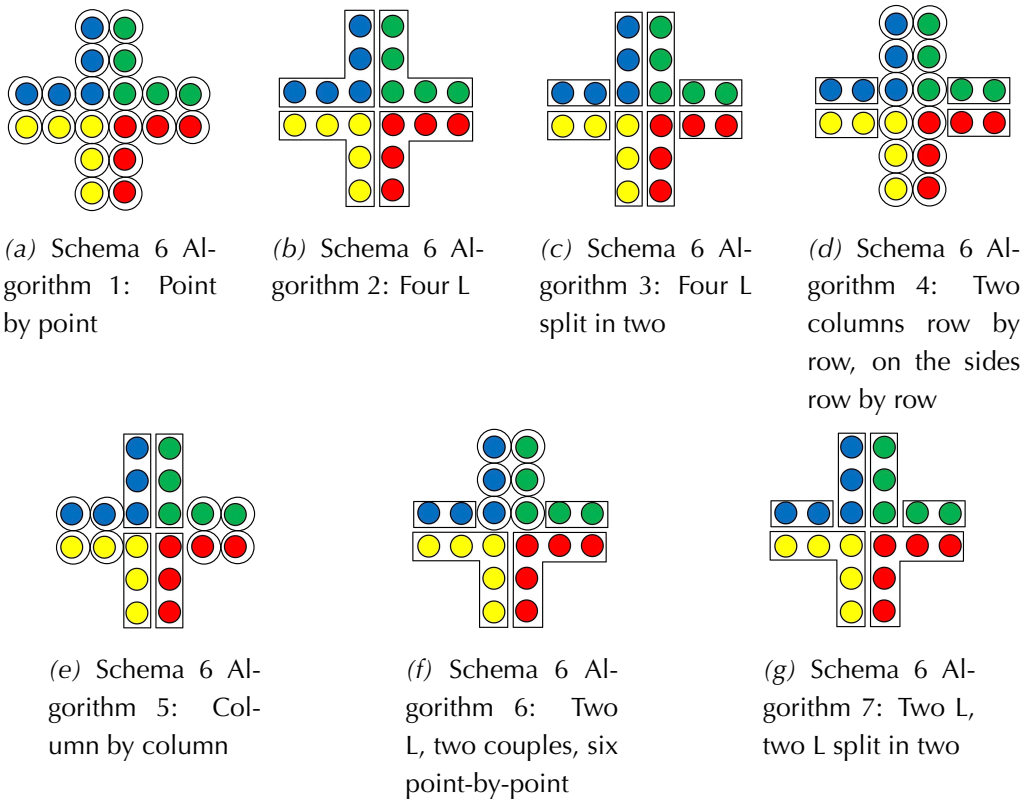
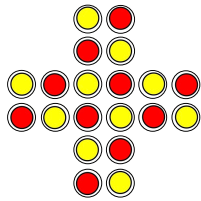
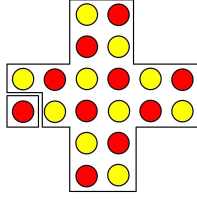


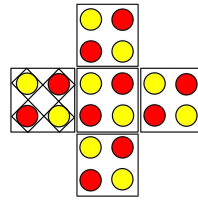
Figure C.6 – Algorithms observed for schema S6.



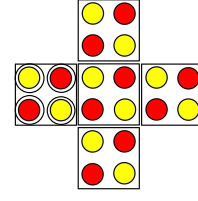
(a) Schema 7 Algorithm 1: Point by point



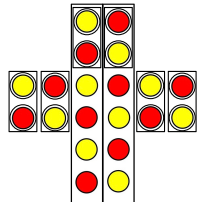
(b) Schema 7 Algorithm 2: Chess-board



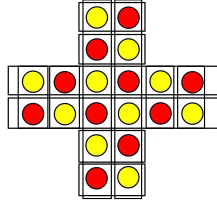
(c) Schema 7 Algorithm 3: Five squares composed of two diagonals



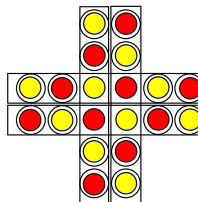
(d) Schema 7 Algorithm 4: Five squares point by point



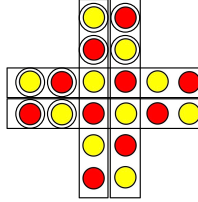
(e) Schema 7 Algorithm 5: Alternated columns



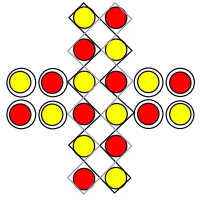
(f) Schema 7 Algorithm 6: Alternated columns and rows (with redundancy)



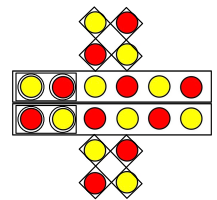
(g) Schema 7 Algorithm 7: Alternated starting from the edges



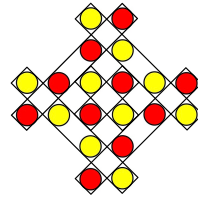
(h) Schema 7 Algorithm 8: Repeated and alternated columns and row



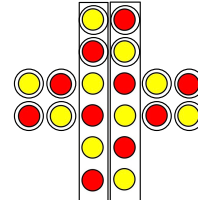
(i) Schema 7 Algorithm 9: Two columns alternated with zig zag, point-by-point on the sides



(j) Schema 7 Algorithm 10: Two rows alternated, diagonals in the squares up and down

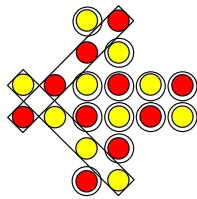


(k) Schema 7 Algorithm 11: Diagonal by diagonal

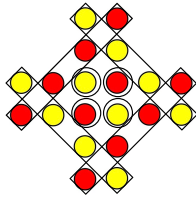


(l) Schema 7 Algorithm 12: Two columns alternated, two squares point-by-point

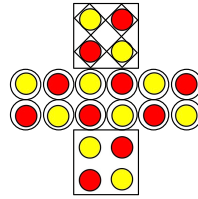
Figure C.7 – Algorithms observed for schema S7. (Continued on the next page).



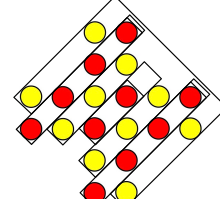
(m) Schema 7 Algorithm 13: Two diagonals, the remaining point-by-point



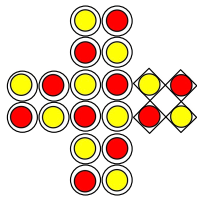
(n) Schema 7 Algorithm 14: Four diagonals, central square point-by-point



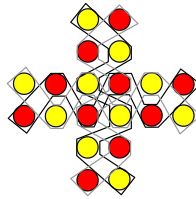
(o) Schema 7 Algorithm 15: Two rows point by point, two squares equals



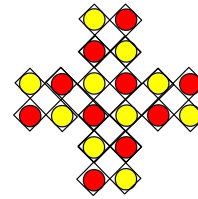
(p) Schema 7 Algorithm 16: Red diagonals, the remaining yellow



(q) Schema 7 Algorithm 17: Point by point, one square composed of diagonals

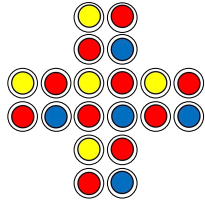


(r) Schema 7 Algorithm 18: Vertical and horizontal zig zag

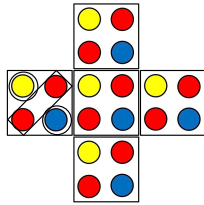


(s) Schema 7 Algorithm 19: Diagonals of two points with intersection (with redundancy)

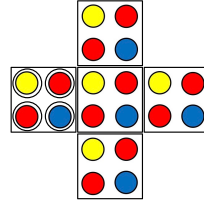
Figure C.7 – Algorithms observed for schema S7. (Continued from the previous page).



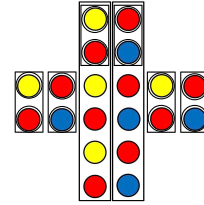
(a) Schema 8 Algorithm 1: Point by point



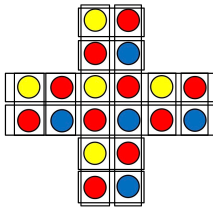
(b) Schema 8 Algorithm 2: Five squares with red diagonal and two points



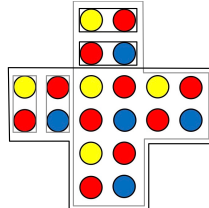
(c) Schema 8 Algorithm 3: Five squares point by point



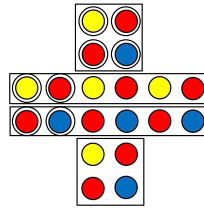
(d) Schema 8 Algorithm 4: Column by column alternated



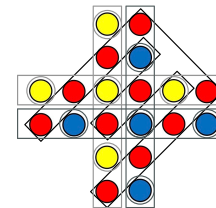
(e) Schema 8 Algorithm 5: Alternated columns and rows (with redundancy)



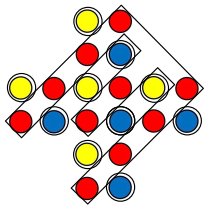
(f) Schema 8 Algorithm 6: Repeated and alternated column by column and row by row



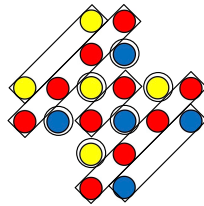
(g) Schema 8 Algorithm 7: Two rows alternated, two identical squares point-by-point



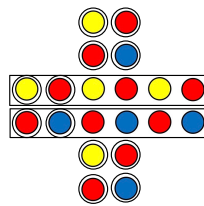
(h) Schema 8 Algorithm 8: Red alternated one yes and one no vertically and horizontally, yellow among reds in the first columns and in the first row, the others blue



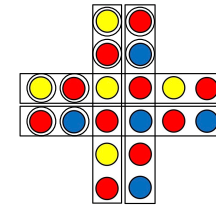
(i) Schema 8 Algorithm 9: Red alternated one yes and one no vertically and horizontally, the remaining point-by-point



(j) Schema 8 Algorithm 10: Diagonal by diagonal

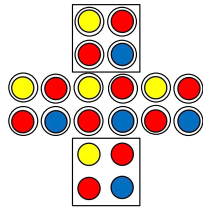


(k) Schema 8 Algorithm 11: Two rows alternated, two squares point-by-point

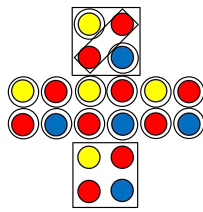


(l) Schema 8 Algorithm 12: Repeated and alternated columns and row

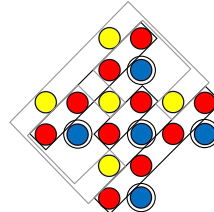
Figure C.8 – Algorithms observed for schema S8. (Continued on the next page).



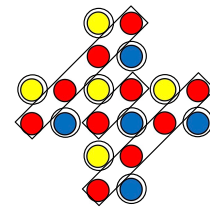
(m) Schema 8 Algorithm 13: Two rows point by point, two identical squares point-by-point



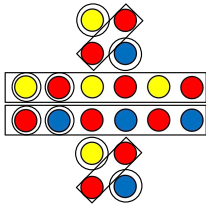
(n) Schema 8 Algorithm 14: Two rows point by point, two identical squares with diagonal



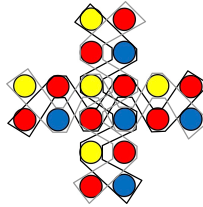
(o) Schema 8 Algorithm 15: Red diagonals, blue point-by-point, remaining yellow



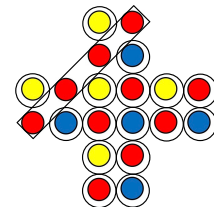
(p) Schema 8 Algorithm 16: Red diagonals, the remaining point-by-point



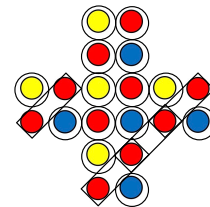
(q) Schema 8 Algorithm 17: Two rows alternated, two squares with diagonals



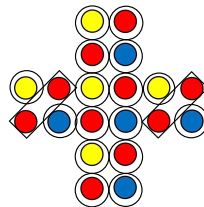
(r) Schema 8 Algorithm 18: Vertical and horizontal zig zag



(s) Schema 8 Algorithm 19: One red diagonal of four points, the remaining point-by-point

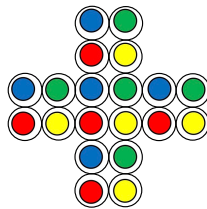


(t) Schema 8 Algorithm 20: Four pairs of red, the remaining point-by-point

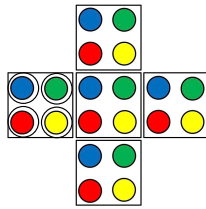


(u) Schema 8 Algorithm 21: Two squares with diagonal, the remaining point-by-point

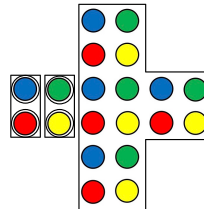
Figure C.8 – Algorithms observed for schema S8. (Continued from the previous page).



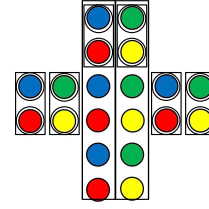
(a) Schema 9 Algorithm 1: Point by point



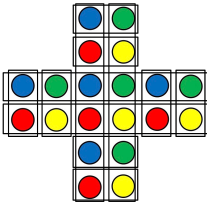
(b) Schema 9 Algorithm 2: Five squares point by point



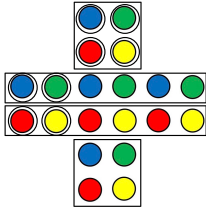
(c) Schema 9 Algorithm 3: Repeated and alternated column by column



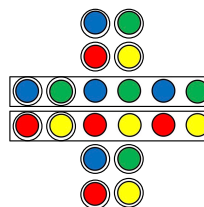
(d) Schema 9 Algorithm 4: Column by column alternated



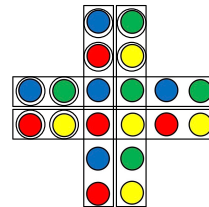
(e) Schema 9 Algorithm 5: Alternated columns and rows (with redundancy)



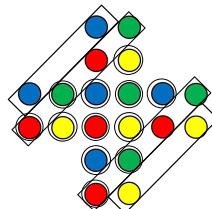
(f) Schema 9 Algorithm 6: Two rows alternated, two identical squares point-by-point



(g) Schema 9 Algorithm 7: Two rows alternated, two squares point-by-point

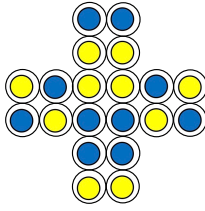


(h) Schema 9 Algorithm 8: Repeated and alternated columns and row

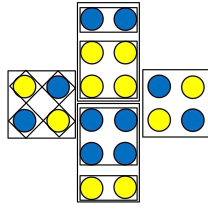


(i) Schema 9 Algorithm 9: Diagonal by diagonal with pairs

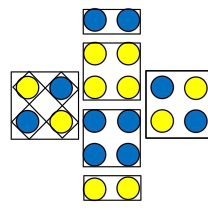
Figure C.9 – Algorithms observed for schema S9.



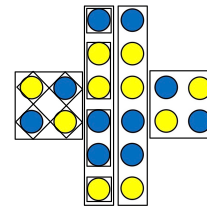
(a) Schema 10 Algorithm 1: Point by point



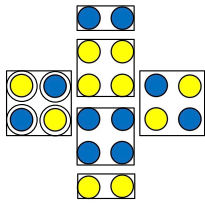
(b) Schema 10 Algorithm 2: Inverted symmetry for the two columns, two symmetric squares on the sides composed by diagonal



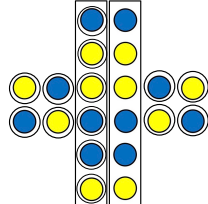
(c) Schema 10 Algorithm 3: Two columns with two rows and two squares, two symmetric squares on the sides composed by diagonal



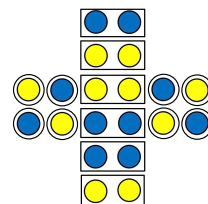
(d) Schema 10 Algorithm 4: Two identical columns with two pairs, two symmetric squares on the sides composed by diagonal



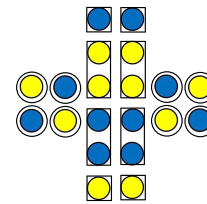
(e) Schema 10 Algorithm 5: Two columns with two rows and two squares, two symmetric squares on the sides point by point



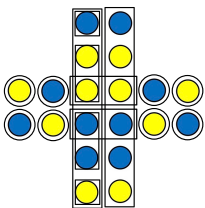
(f) Schema 10 Algorithm 6: Two identical columns point-by-point, two sides point-by-point



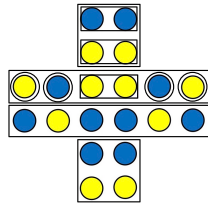
(g) Schema 10 Algorithm 7: Two columns row by row, two sides point-by-point



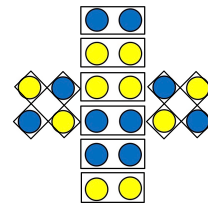
(h) Schema 10 Algorithm 8: Column by column



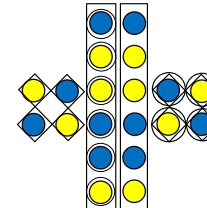
(i) Schema 10 Algorithm 9: Two identical columns with two pairs, two rows with pairs (with redundancy)



(j) Schema 10 Algorithm 10: Two identical squares up and down with pairs, two symmetric rows with pair of points

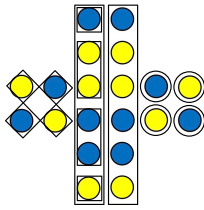


(k) Schema 10 Algorithm 11: Two columns row by row, double diagonal on both sides

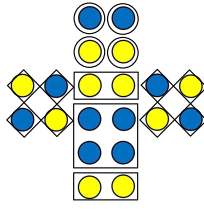


(l) Schema 10 Algorithm 12: Two identical columns point-by-point, double diagonal on both sides (one side redundant point by point)

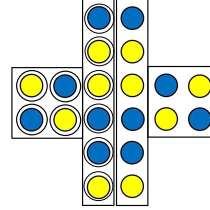
Figure C.10 – Algorithms observed for schema S10. (Continued on the next page).



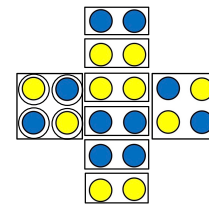
(m) Schema 10 Algorithm 13: Two columns with two rows and two squares, one side diagonals, other side point-by-point



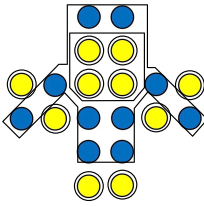
(n) Schema 10 Algorithm 14: In the central column a square point-by-point, then a row then square and row, double diagonal on both sides



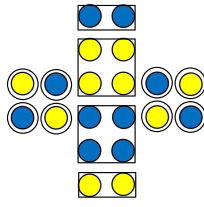
(o) Schema 10 Algorithm 15: Two identical columns point-by-point, two symmetric squares point-by-point



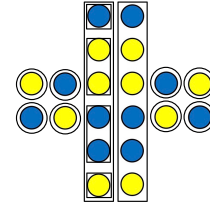
(p) Schema 10 Algorithm 16: Two columns row by row, two symmetric squares point-by-point



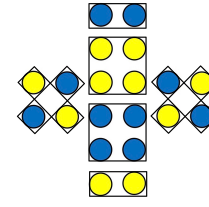
(q) Schema 10 Algorithm 17: One colour point by point, the remaining with the other colour



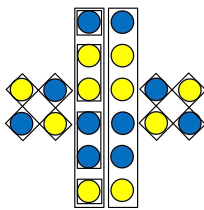
(r) Schema 10 Algorithm 18: Two columns with two rows and two squares, two squares on the sides point-by-point



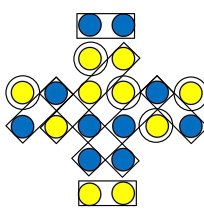
(s) Schema 10 Algorithm 19: Two identical columns with two pairs, two squares on the sides point-by-point



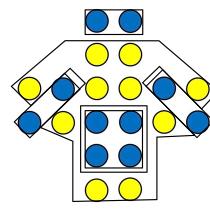
(t) Schema 10 Algorithm 20: Two columns with two rows and two squares, double diagonal on both sides



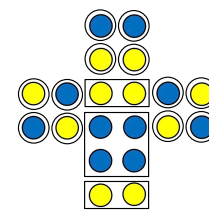
(u) Schema 10 Algorithm 21: Two identical columns with two pairs, double diagonal on both sides



(v) Schema 10 Algorithm 22: Blue in pairs and

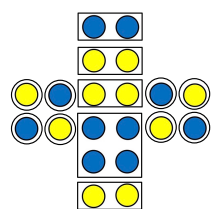


(w) Schema 10 Algorithm 23: One square blue, other blue in pairs, the remaining yellow



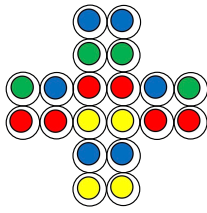
(x) Schema 10 Algorithm 24: In the central column a square point-by-point, two rows and a square, two sides point-by-point

Figure C.10 – Algorithms observed for schema S10. (Continued from the previous page).

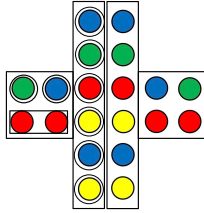


(y) Schema 10
Algorithm 25:
In the central
column a square
and remaining by
rows, two sides
point-by-point

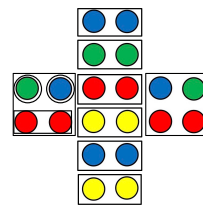
Figure C.10 – Algorithms observed for schema S10. (Continued from the previous page).



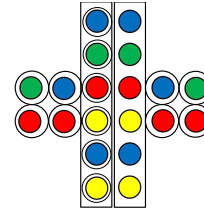
(a) Schema 11 Algorithm 1: Point by point



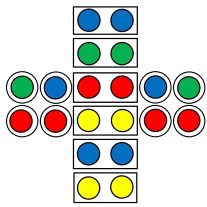
(b) Schema 11 Algorithm 2: Two identical columns point-by-point, on one side a red line and two points, mirrored on the other side



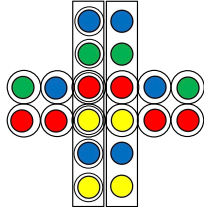
(c) Schema 11 Algorithm 3: Two columns row by row, on one side a red line and two points, mirrored on the other side



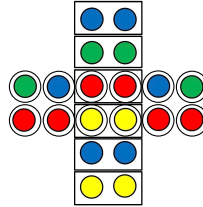
(d) Schema 11 Algorithm 4: Two identical columns point-by-point, on the sided point-by-point



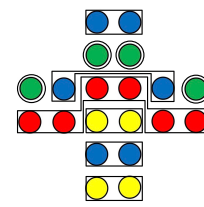
(e) Schema 11 Algorithm 5: Two columns row by row, on the sided point-by-point



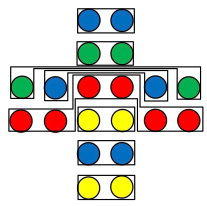
(f) Schema 11 Algorithm 6: Two identical columns point-by-point, two rows point-by-point (with redundancy)



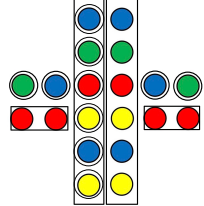
(g) Schema 11 Algorithm 7: Two columns row by row, two rows point-by-point (with redundancy)



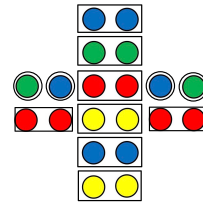
(h) Schema 11 Algorithm 8: Blue and yellow in pairs, green point-by-point, remaining in red



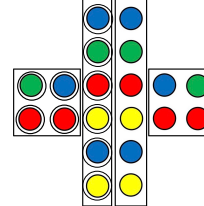
(i) Schema 11 Algorithm 9: Blue, yellow and green in pairs, the remaining red



(j) Schema 11 Algorithm 10: Two identical columns point-by-point, two pairs of red, the remaining point by point

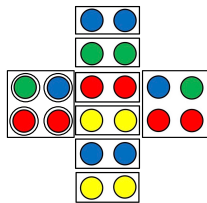


(k) Schema 11 Algorithm 11: Two columns row by row, two pairs of red, the remaining point-by-point

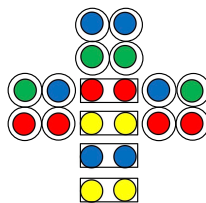


(l) Schema 11 Algorithm 12: Two identical columns point-by-point, on one side point-by-point, mirrored on the other side

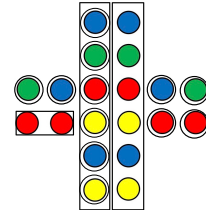
Figure C.11 – Algorithms observed for schema S11. (Continued on the next page).



(m) Schema 11
Algorithm 13: Two
columns with 4
pairs, the remain-
ing point-by-point

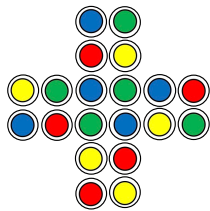


(n) Schema 11
Algorithm 14: Two
identical columns
point-by-point,
one pair of red, the
remaining point
by point

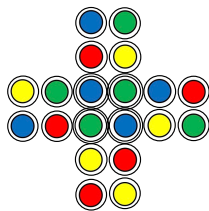


(o) Schema 11 Al-
gorithm 15: Two
columns row by
row, one pair of
red, the remaining
point-by-point

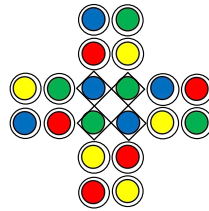
Figure C.11 – Algorithms observed for schema S11. (Continued from the previous page).



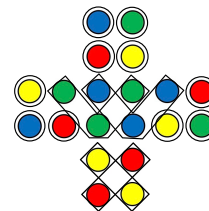
(a) Schema 12 Algorithm 1: Point by point



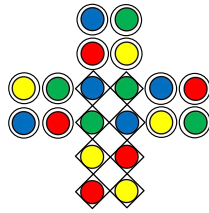
(b) Schema 12 Algorithm 2: Point by point (with redundancy)



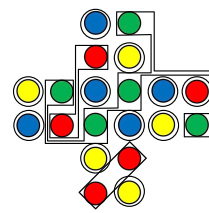
(c) Schema 12 Algorithm 3: Point by point except one square with diagonals



(d) Schema 12 Algorithm 4: L of three green, L of three blue, a square with diagonals, the remaining point by point



(e) Schema 12 Algorithm 5: Point by point except two squares with diagonals



(f) Schema 12 Algorithm 6: Blue and yellow point-by-point, red in pairs, the remaining green

Figure C.12 – Algorithms observed for schema S12.

C.3 Algorithmic and interaction strategies by schema

This section presents an overview of the algorithmic and interaction strategies developed by students for each schema of the unplugged CAT. By analysing the approaches, we aim to illustrate how students of different age groups engaged with the task and adapted their problem-solving methods. This detailed examination reveals the diversity of strategies employed and highlights patterns specific to each schema, offering insights into the cognitive processes underlying AT.

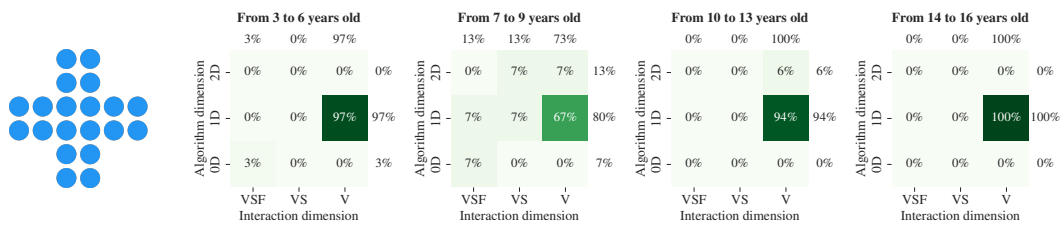


Figure C.13 – Algorithmic and interaction strategies across age for S1.

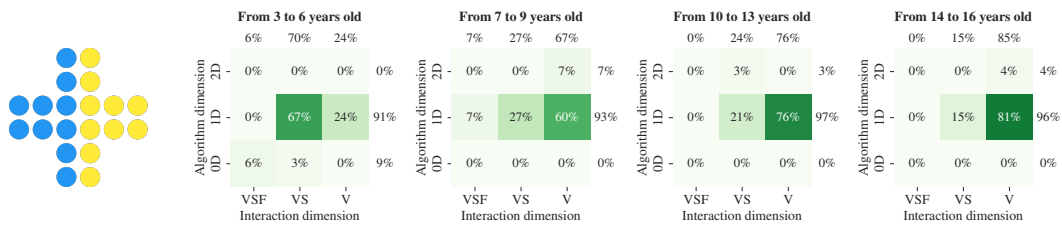


Figure C.14 – Algorithmic and interaction strategies across age for S2.

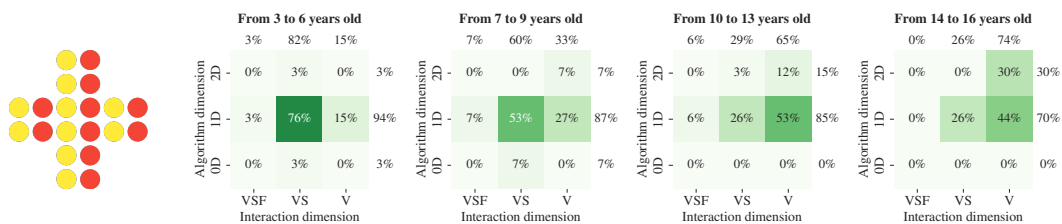


Figure C.15 – Algorithmic and interaction strategies across age for S3.

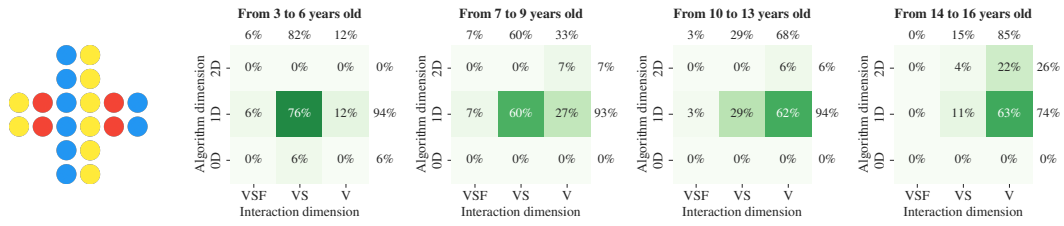


Figure C.16 – Algorithmic and interaction strategies across age for S4.

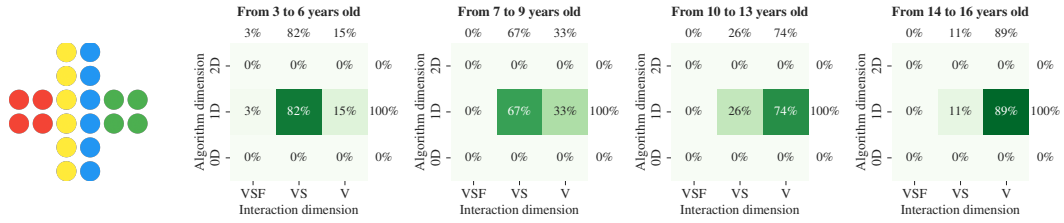


Figure C.17 – Algorithmic and interaction strategies across age for S5.

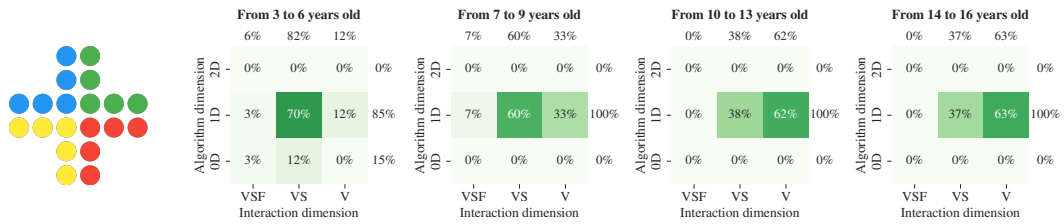


Figure C.18 – Algorithmic and interaction strategies across age for S6.

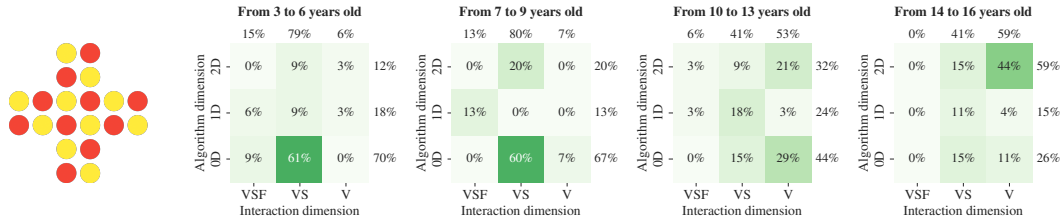


Figure C.19 – Algorithmic and interaction strategies across age for S7.

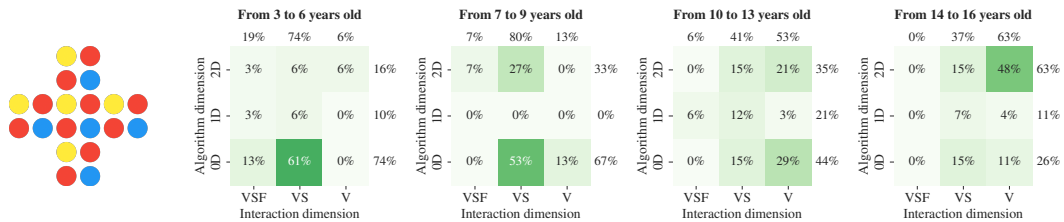


Figure C.20 – Algorithmic and interaction strategies across age for S8.

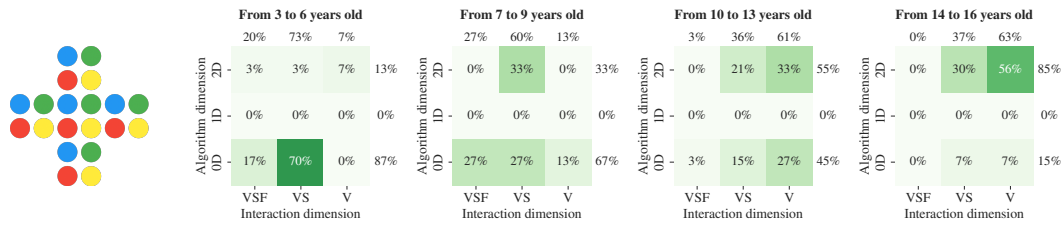


Figure C.21 – Algorithmic and interaction strategies across age for S9.

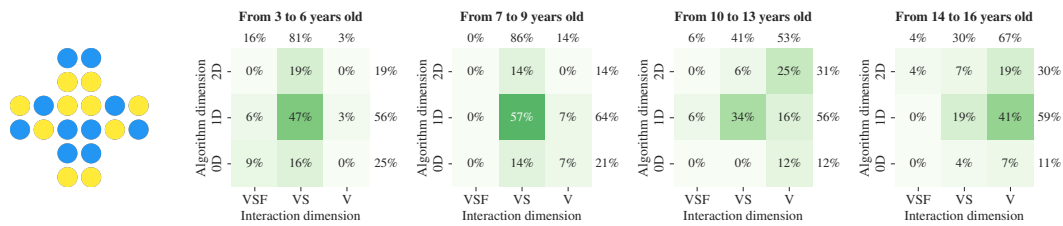


Figure C.22 – Algorithmic and interaction strategies across age for S10.

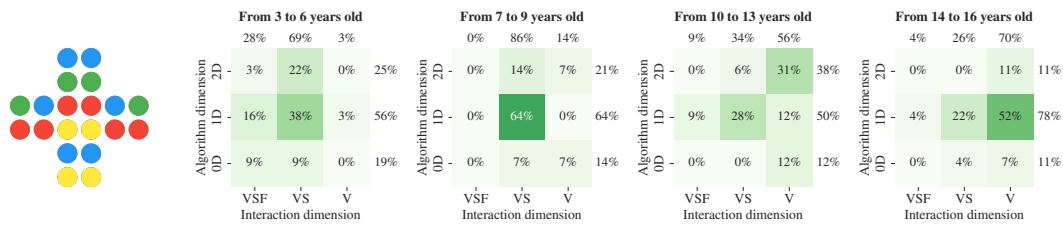


Figure C.23 – Algorithmic and interaction strategies across age for S11.



Figure C.24 – Algorithmic and interaction strategies across age for S12.

Appendix D

Pilot study with the virtual CAT

This appendix presents the details of the pilot study conducted with the virtual CAT, in particular we included: (i) the final application's user interface; (ii) illustrations of the performance for each schema, specifically showing the development of algorithmic and interaction strategies.

D.1 Screens of the final application

This section provides screenshots of the final application from the pilot study, including all key stages of the virtual CAT assessment. The images cover the initial language selection for the test, the choice between training or validation modes, data entry screens, and the main testing interface with its three different interaction modes. Additionally, it includes visuals of the final results dashboard and the survey screen, where participants provided feedback on the application and its activities. These screenshots offer a detailed overview of the user experience, highlighting the flow and layout of the virtual environment.

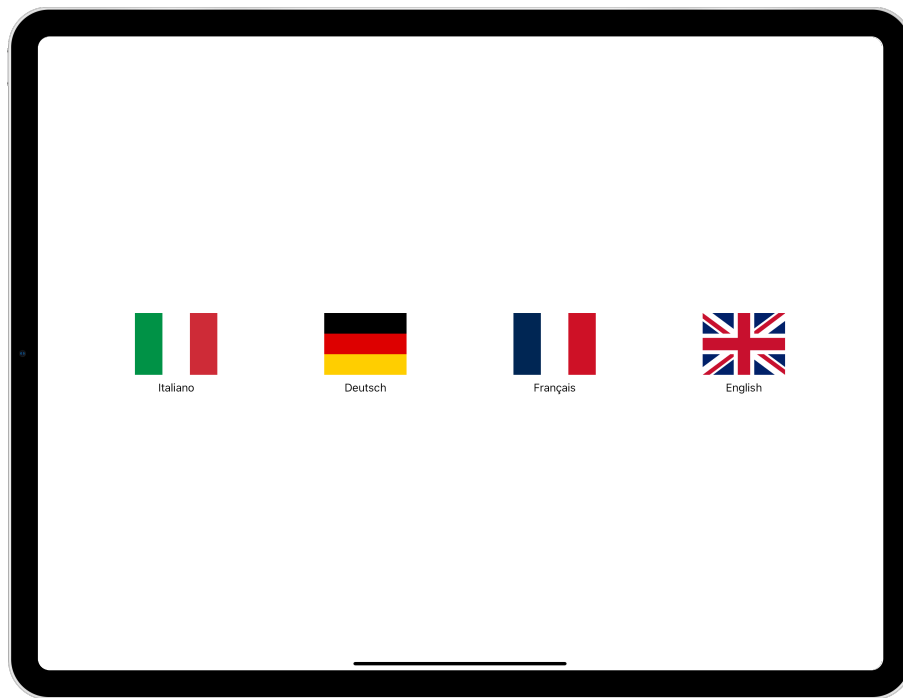


Figure D.1 – Language selection.

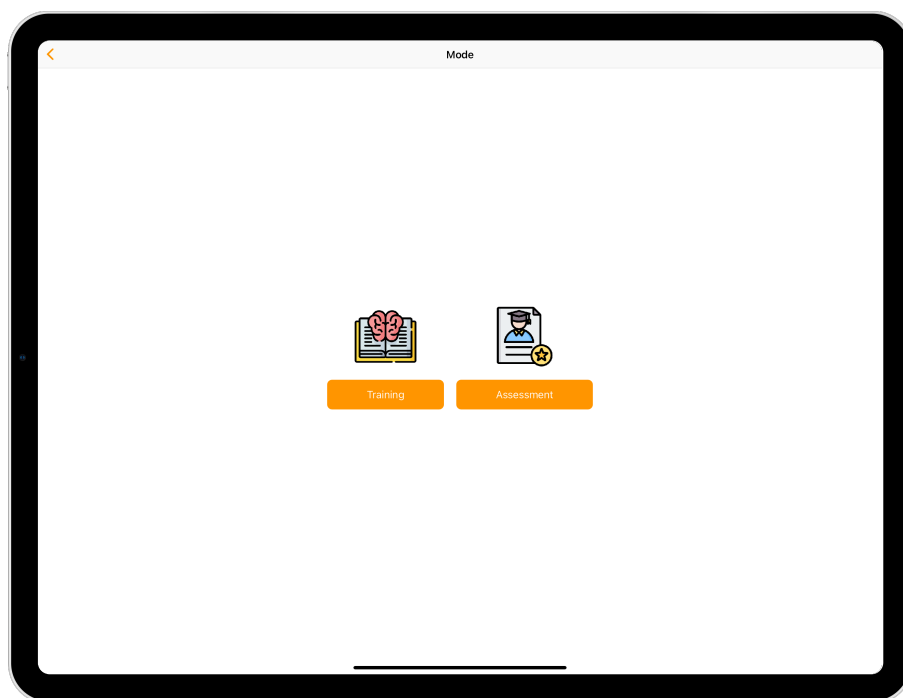


Figure D.2 – Module selection.

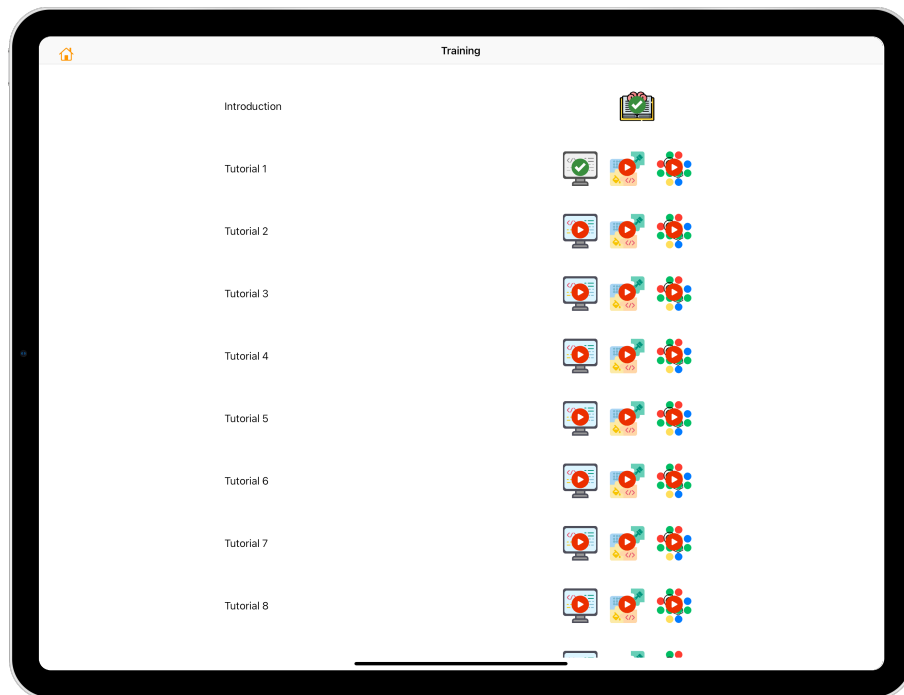
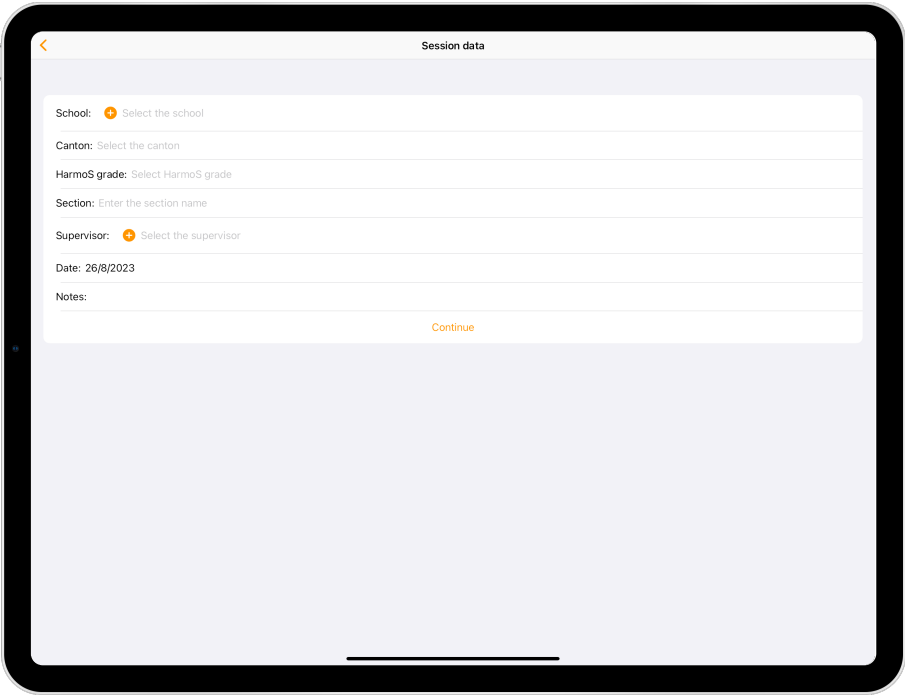


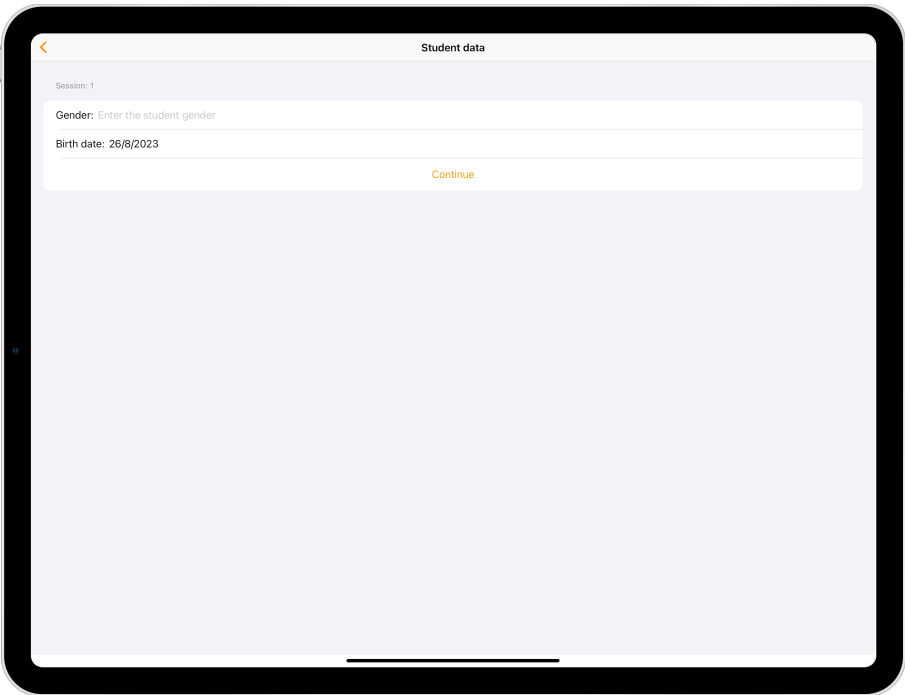
Figure D.3 – Training module.

An introductory video about the application is provided on the training screen, followed by a series of explanatory videos for all practice tasks in each interface. After watching the video, users can attempt to solve the schema using the provided instructions. When a schema is successfully solved, the video icon is marked with a green checkmark.



The image shows a mobile application screen titled "Session data". It features a light gray header with a back arrow on the left and the title "Session data" on the right. Below the header is a white form with several input fields: "School:" with a dropdown arrow and the text "Select the school"; "Canton:" with the text "Select the canton"; "HarmoS grade:" with the text "Select HarmoS grade"; "Section:" with the text "Enter the section name"; "Supervisor:" with a dropdown arrow and the text "Select the supervisor"; "Date:" with the value "26/8/2023"; and "Notes:" with a large text area. An orange "Continue" button is positioned at the bottom right of the form. The entire screen is framed by a black border, and a horizontal line is visible at the bottom, indicating a home indicator.

Figure D.4 – Session form in the validation module.



The image shows a mobile application screen titled "Student data". It features a light gray header with a back arrow on the left and the title "Student data" on the right. Below the header is a white form with two input fields: "Gender:" with the text "Enter the student gender" and "Birth date:" with the value "26/8/2023". An orange "Continue" button is positioned at the bottom right of the form. The entire screen is framed by a black border, and a horizontal line is visible at the bottom, indicating a home indicator.

Figure D.5 – Student form in the validation module.

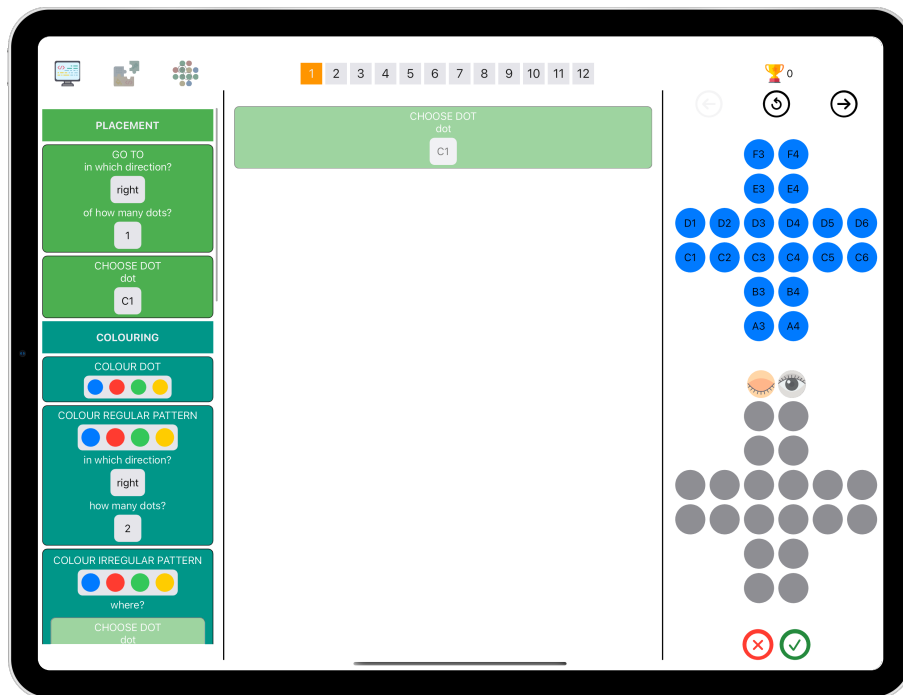


Figure D.6 – CAT visual programming interface (CAT-VPI) with textual commands.

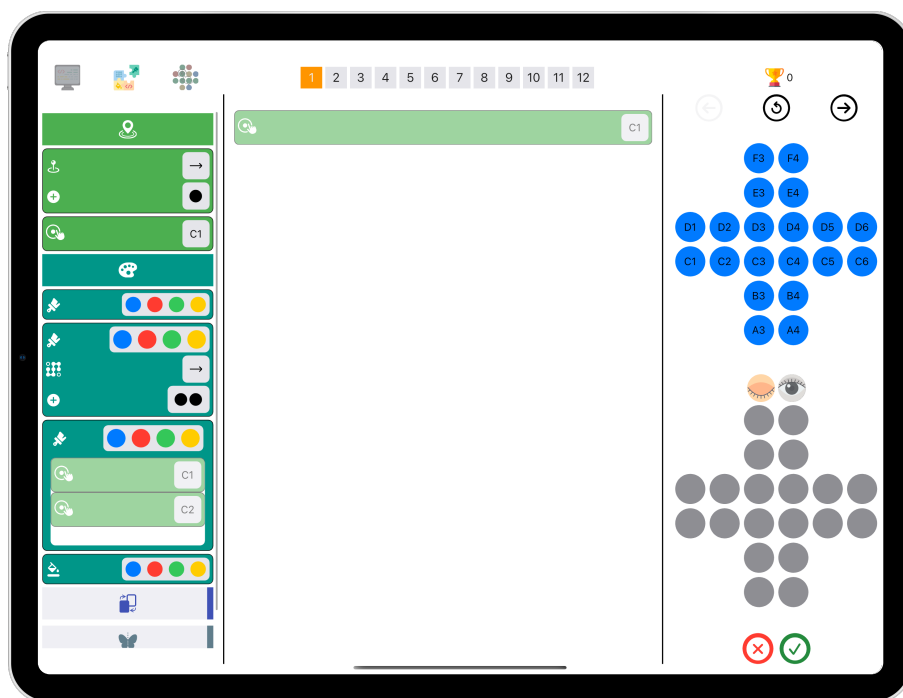


Figure D.7 – CAT visual programming interface (CAT-VPI) with symbolic commands.

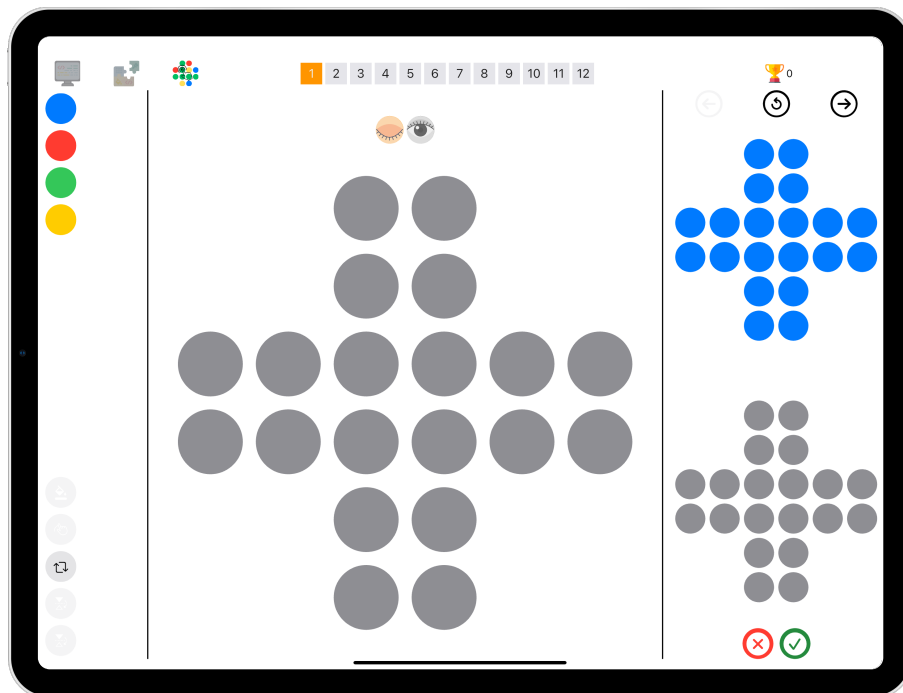


Figure D.8 – CAT gesture interface (CAT-GI).

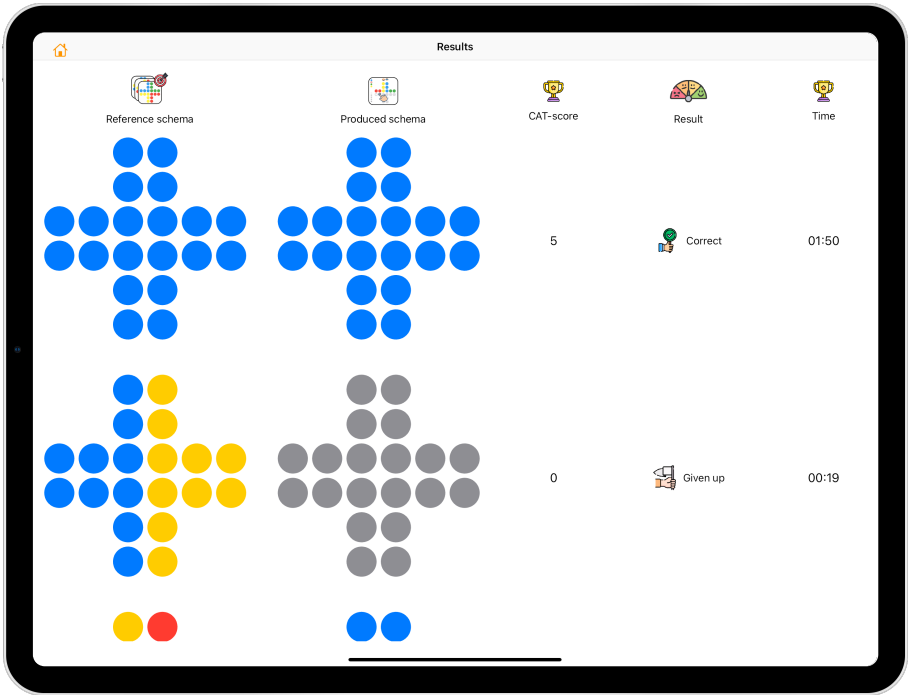


Figure D.9 – Results dashboard. It comprehensively summarises pupils’ performance across all schemas. This dashboard includes a visual representation of reference schemas alongside those resulting from student instructions, the pupil’s score, an indication of whether each schema was completed correctly, incorrectly, or skipped, and the time taken to complete the schema.

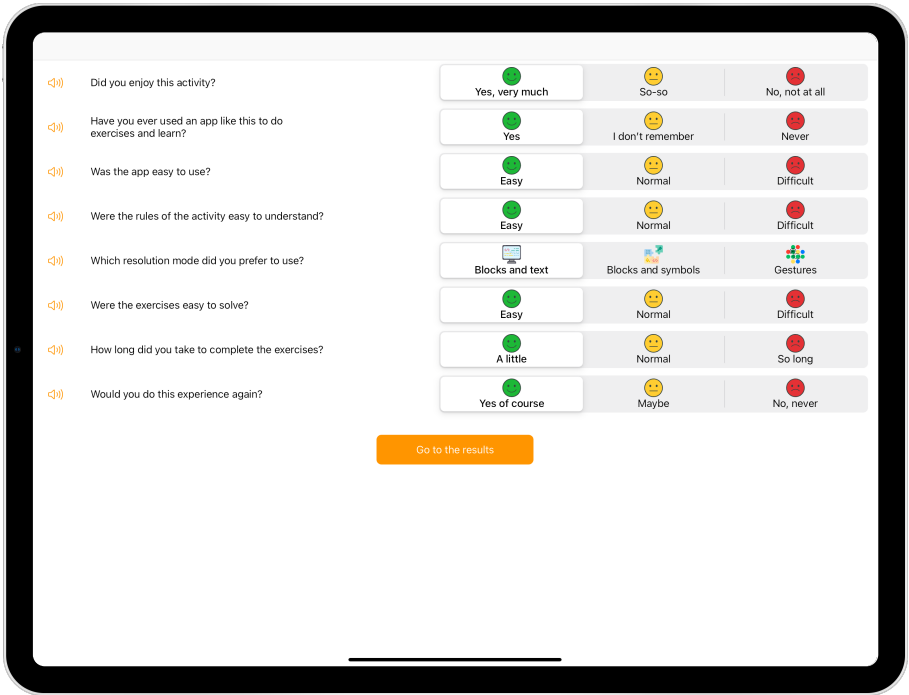


Figure D.10 – Pupil feedback survey. The voice-assisted questions evaluate user interactions with the app. Each question is accompanied by three distinct emoticon-style response options: a contented smiling face, a neutral face, and a discontented frowning face. A concluding button invites users to view aggregated results.

D.2 Algorithmic and interaction strategies by schema

This section presents an overview of the algorithmic and interaction strategies developed by students for each schema of the virtual CAT. By analysing the approaches, we aim to illustrate how students of different age groups engaged with the task and adapted their problem-solving methods. This detailed examination reveals the diversity of strategies employed and highlights patterns specific to each schema, offering insights into the cognitive processes underlying AT.

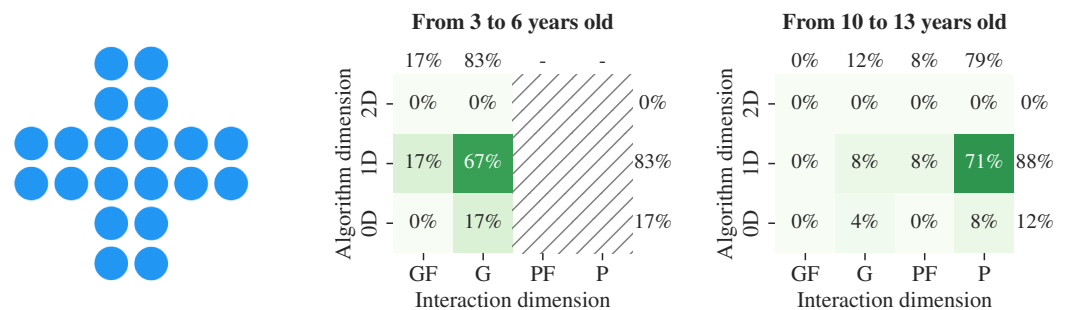


Figure D.11 – Algorithmic and interaction strategies across age for S1.

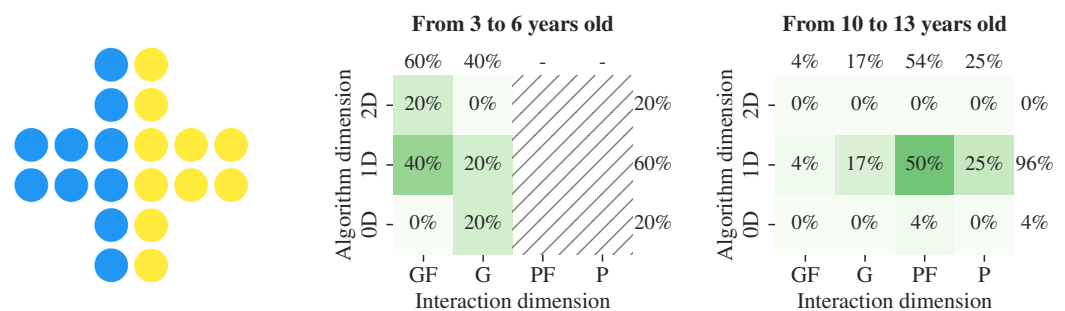


Figure D.12 – Algorithmic and interaction strategies across age for S2.

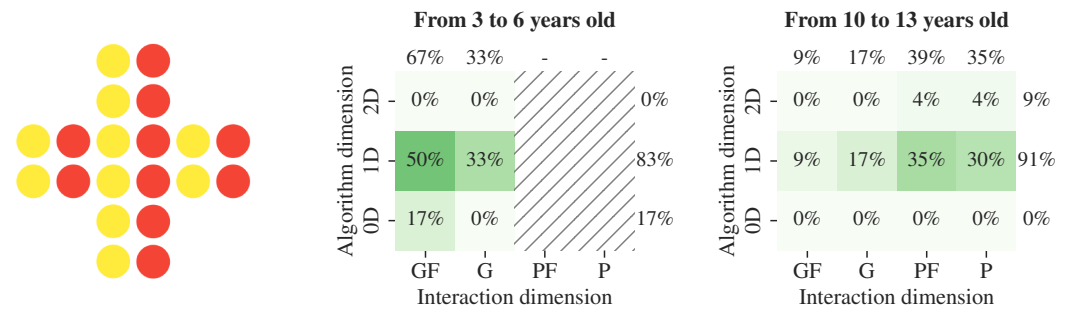


Figure D.13 – Algorithmic and interaction strategies across age for S3.

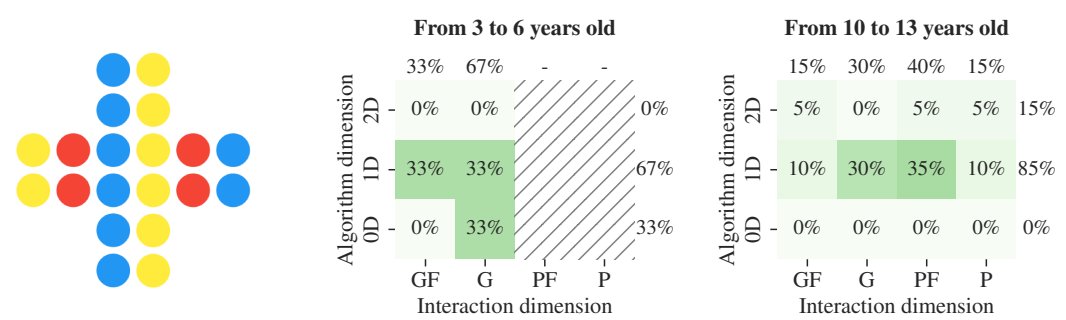


Figure D.14 – Algorithmic and interaction strategies across age for S4.

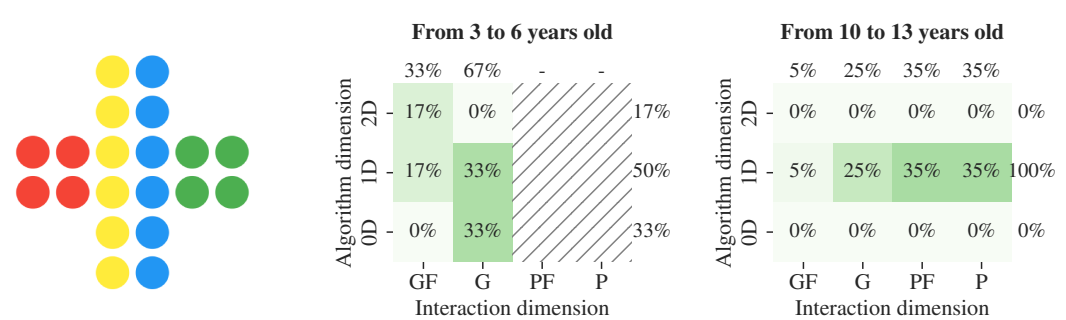


Figure D.15 – Algorithmic and interaction strategies across age for S5.

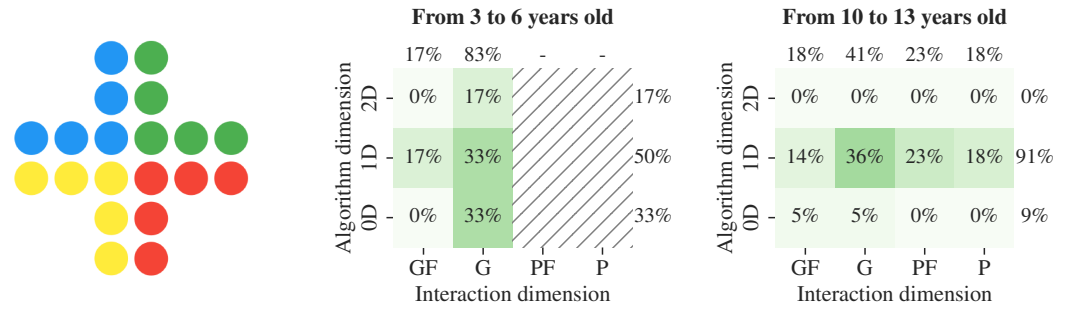


Figure D.16 – Algorithmic and interaction strategies across age for S6.

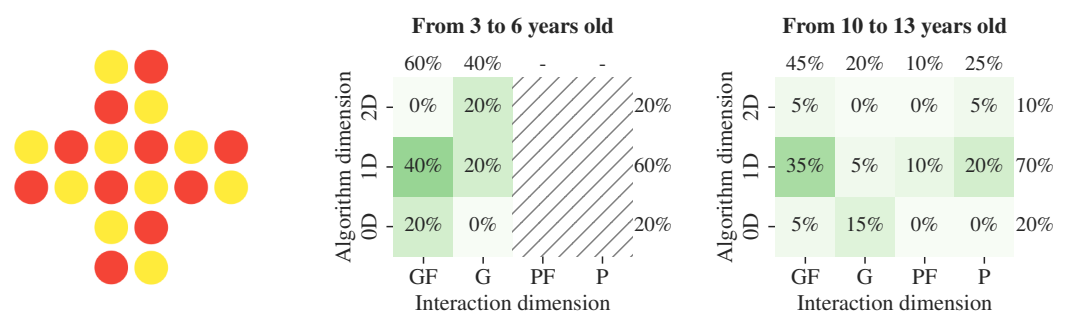


Figure D.17 – Algorithmic and interaction strategies across age for S7.

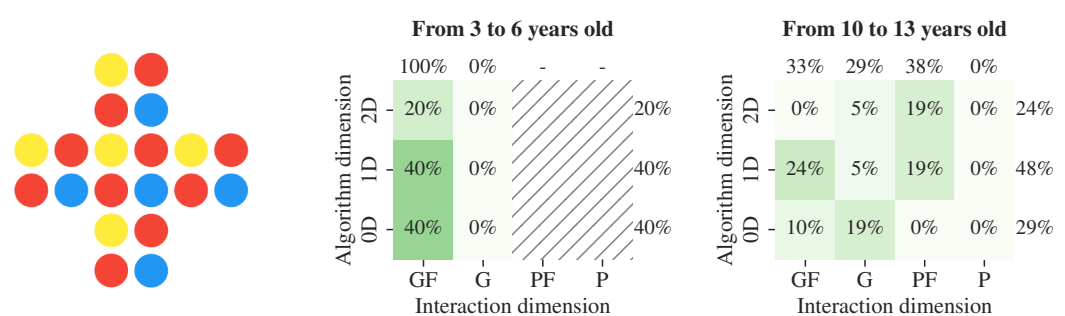


Figure D.18 – Algorithmic and interaction strategies across age for S8.

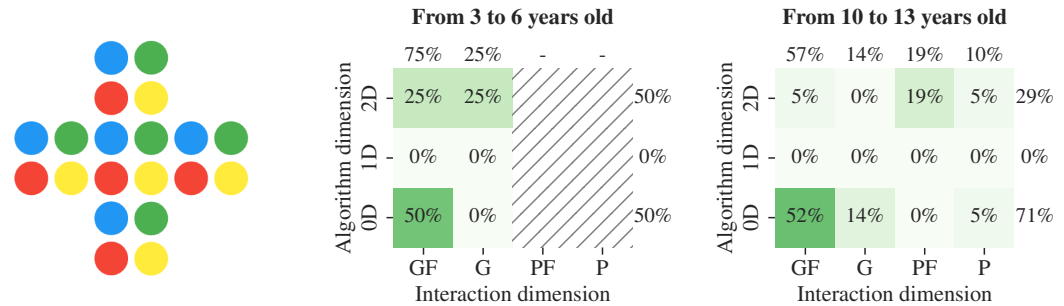


Figure D.19 – Algorithmic and interaction strategies across age for S9.

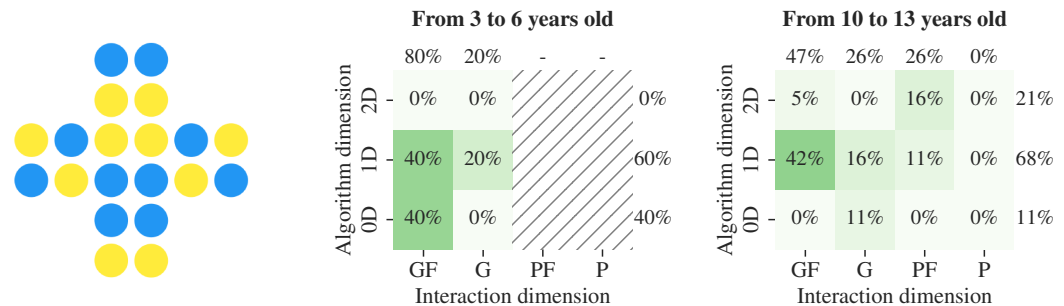


Figure D.20 – Algorithmic and interaction strategies across age for S10.

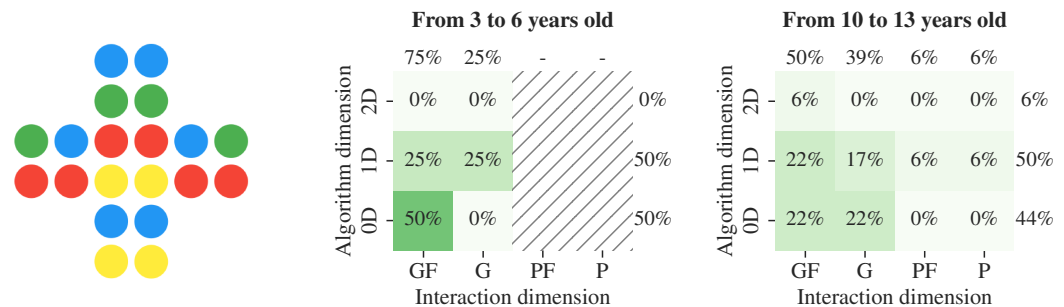


Figure D.21 – Algorithmic and interaction strategies across age for S11.

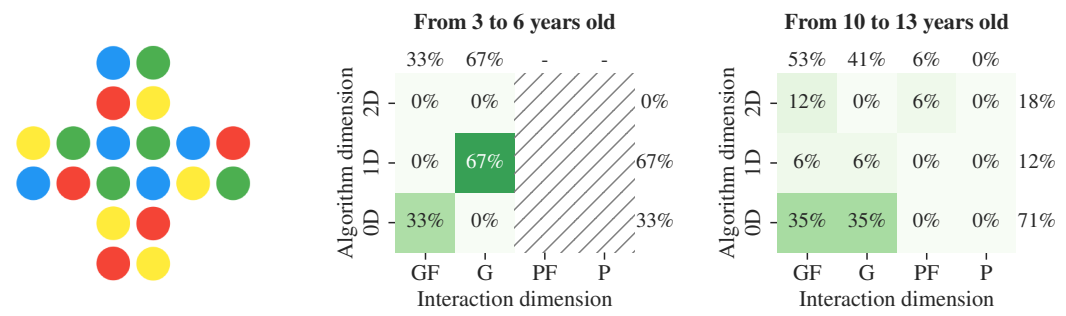


Figure D.22 – Algorithmic and interaction strategies across age for S12.

Bibliography

- [1] Acara (2015). Digital technologies foundation to year 10 scope and sequence. Australian Curriculum, Assessment and Reporting Authority. https://docs.acara.edu.au/resources/Digital_Technologies_-_Sequence_of_content.pdf.
- [2] Adorni, G. (2023). Dataset for algorithmic thinking skills assessment: Results from the virtual CAT pilot study in Swiss compulsory education (1.0.0). Zenodo Dataset. <https://doi.org/10.5281/zenodo.10018292>.
- [3] Adorni, G. (2024a). Dataset for algorithmic thinking skills assessment: Results from the virtual CAT large-scale study in Swiss compulsory education (1.0.0). Zenodo Dataset. <https://doi.org/10.5281/zenodo.10912339>.
- [4] Adorni, G. (2024b). Virtual CAT Algorithmic Thinking Assessment: Data Analysis Procedures. Zenodo Software. <https://doi.org/10.5281/zenodo.12805318>.
- [5] Adorni, G., Artico, I., Piatti, A., Lutz, E., Gambardella, L. M., Negrini, L., Mondada, F., and Assaf, D. (2024a). Development of algorithmic thinking skills in K-12 education: A comparative study of unplugged and digital assessment instruments. *Computers in Human Behavior Reports*, 15:100466. <https://doi.org/10.1016/j.chbr.2024.100466>.
- [6] Adorni, G. and Bonesana, C. (2023). BN-based Learning Networks with Noisy Gates (1.0.0). Zenodo Software. <https://doi.org/10.5281/zenodo.10958613>.
- [7] Adorni, G. and Karpenko, V. (2023a). virtual CAT data infrastructure (1.0.0). Zenodo Software. <https://doi.org/10.5281/zenodo.10015011>.
- [8] Adorni, G. and Karpenko, V. (2023b). virtual CAT programming language interpreter (1.0.0). Zenodo Software. <https://doi.org/10.5281/zenodo.10016535>.

- [9] Adorni, G., Mangili, F., Piatti, A., Bonesana, C., and Antonucci, A. (2023a). Rubric-based Learner Modelling via Noisy Gates Bayesian Networks for Computational Thinking Skills Assessment. *Journal of Communications Software and Systems*, 19(1):52–64. <http://doi.org/10.24138/jcomss-2022-0169>.
- [10] Adorni, G. and Piatti, A. (2024). Designing the virtual CAT: A digital tool for algorithmic thinking assessment in compulsory education. To be submitted to the International Journal of Child-Computer Interaction. <https://doi.org/10.48550/arXiv.2408.01263>.
- [11] Adorni, G., Piatti, A., Bumbacher, E., Negrini, L., Mondada, F., Assaf, D., Mangili, F., and Gambardella, L. (2025). FADE-CTP: A Framework for the Analysis and Design of Educational Computational Thinking Problems. *Technology, Knowledge and Learning*. [10.1007/s10758-025-09833-x](https://doi.org/10.1007/s10758-025-09833-x).
- [12] Adorni, G., Piatti, S., and Karpenko, V. (2023b). virtual CAT: An app for algorithmic thinking assessment within Swiss compulsory education (1.0.1). Zenodo Software. <https://doi.org/10.5281/zenodo.10027851>.
- [13] Adorni, G., Piatti, S., and Karpenko, V. (2024b). Virtual CAT: A multi-interface educational platform for algorithmic thinking assessment. *SoftwareX*, 27:101737. <https://doi.org/10.1016/j.softx.2024.101737>.
- [14] Aebi-Müller, R. E., Blatter, I., Brigger, J., Constable, E. C., Eglin, N., Hoffmeyer, P., Lautenschütz, C., Lienhard, A., Pirinoli, C., Röthlisberger, M., and Spycher, K. M. (2021). Code of conduct for scientific integrity. <https://doi.org/10.5281/ZENODO.4707560>.
- [15] Ahmad, N. A. N. and Hussaini, M. (2021). A usability testing of a higher education mobile application among postgraduate and undergraduate students. *International Journal of Interactive Mobile Technologies (IJIM)*, 15(09):88. <https://doi.org/10.3991/ijim.v15i09.19943>.
- [16] Ahmad, W., Shahnawazuddin, S., Kathania, H., Pradhan, G., and Samaddar, A. (2017). Improving children’s speech recognition through explicit pitch scaling based on iterative spectrogram inversion. In *Interspeech 2017*. ISCA. <https://doi.org/10.21437/interspeech.2017-302>.
- [17] Aimin Zhou, C. X. (2024). Svcgan: Speaker voice conversion generative adversarial network for children’s speech conversion and recognition. *Journal of Electrical Systems*, 20(3s):2182–2196. <https://doi.org/10.52783/jes.1841>.

- [18] Amini, R., Stolz, L. A., Kartchner, J. Z., Thompson, M., Stea, N., Hawbaker, N., Joshi, R., and Adhikari, S. (2016). Bedside echo for chest pain: An algorithm for education and assessment. *Advances in Medical Education and Practice*, page 293. <https://doi.org/10.2147/amep.s103083>.
- [19] Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., and Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, 19(3):47–57. <https://www.jstor.org/stable/pdf/jeductechsoci.19.3.47.pdf>.
- [20] Antonucci, A., Mangili, F., Bonesana, C., and Adorni, G. (2021). A New Score for Adaptive Tests in Bayesian and Credal Networks. In *Vejnarová, J., Wilson, N. (eds) Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 12897, pages 399–412. Springer, Cham. ECSQARU 2021. Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-030-86772-0_29.
- [21] Antonucci, A., Mangili, F., Bonesana, C., and Adorni, G. (2022). Intelligent Tutoring Systems by Bayesian Nets with Noisy Gates. *The International FLAIRS Conference Proceedings*, 35. <https://doi.org/10.32473/flairs.v35i.130692>.
- [22] Ardito, G., Czerkawski, B., and Scollins, L. (2020). Learning computational thinking together: Effects of gender differences in collaborative middle school robotics program. *TechTrends*, 64(3):373–387. <https://doi.org/10.1007/s11528-019-00461-8>.
- [23] Atmatzidou, S. and Demetriadis, S. (2016). Advancing students’ computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75:661–670. <https://doi.org/10.1016/j.robot.2015.10.008>.
- [24] Baecker, R. M. (2014). *Readings in Human-Computer Interaction: toward the year 2000*. Elsevier.
- [25] Barr, V. and Stephenson, C. (2011). Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1):48–54. <https://doi.org/10.1145/1929887.1929905>.
- [26] Bartlett, M. S. (1937). Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 160(901):268–282. <https://doi.org/10.1098/rspa.1937.0109>.

- [27] Bell, T., Alexander, J., Freeman, I., and Grimley, M. (2009). Computer Science Unplugged: school students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13.
- [28] Bell, T. and Vahrenhold, J. (2018). CS Unplugged—How Is It Used, and Does It Work? In *Adventures between lower bounds and higher altitudes*, pages 497–521. Springer International Publishing. https://doi.org/10.1007/978-3-319-98355-4_29.
- [29] Bellino, A. and Herskovic, V. (2023). Protobject as a tool for teaching computational thinking to designers: student perceptions on usability. In *Proceedings of the 15th Biannual Conference of the Italian SIGCHI Chapter*, CHIItaly 2023. ACM. <https://doi.org/10.1145/3605390.3605401>.
- [30] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300. <http://doi.org/10.1111/j.2517-6161.1995.tb02031.x>.
- [31] Bennett, C. (2006). Keeping up with the kids. *Young Consumers*, 7(3):28–32. <https://doi.org/10.1108/17473610610705345>.
- [32] Benoit, L., Lehalle, H., and Jouen, F. (2004). Do young children acquire number words through subitizing or counting? *Cognitive Development*, 19(3):291–307. <https://doi.org/10.1016/j.cogdev.2004.03.005>.
- [33] Bers, M. U., Strawhacker, A., and Sullivan, A. (2022). The state of the field of computational thinking in early childhood education. OECD Education Working Papers 274, OECD Publishing. <https://doi.org/10.1787/3354387a-en>.
- [34] Beyer, S. (2014). Why are women underrepresented in computer science? gender differences in stereotypes, self-efficacy, values, and interests and predictors of future cs course-taking and grades. *Computer Science Education*, 24(2–3):153–192. <http://doi.org/10.1080/08993408.2014.963363>.
- [35] Bland, J. M. and Altman, D. G. (1995). Statistics notes: Multiple significance tests: the bonferroni method. *BMJ*, 310(6973):170–170. <http://dx.doi.org/10.1136/bmj.310.6973.170>.
- [36] Bloom, B. S. et al. (1956). Taxonomy of. *Educational Objectives*.

- [37] Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., and Punie, Y. (2016). Developing computational thinking: Approaches and orientations in k-12 education. In *EdMedia+ Innovate Learning*, pages 13–18. Association for the Advancement of Computing in Education (AACE). <https://www.learntechlib.org/p/172925/>.
- [38] Bocconi, S., Chiocciariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M. A., Jasutė, E., Malagoli, C., et al. (2022). Reviewing Computational Thinking in Compulsory Education. Technical report, Joint Research Centre (Seville site). <https://doi.org/10.2760/126955>.
- [39] Bogdan, R. and Biklen, S. K. (1997). *Qualitative research for education*. Allyn & Bacon Boston, MA. http://math.buffalostate.edu/dwilson/MED595/Qualitative_intro.pdf.
- [40] Borgers, N., de Leeuw, E., and Hox, J. (2000). Children as respondents in survey research: Cognitive development and response quality 1. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 66(1):60–75. <https://doi.org/10.1177/075910630006600106>.
- [41] Brackmann, C. P., Barone, D., Casali, A., Boucinha, R., and Muñoz-Hernandez, S. (2016). Computational thinking: Panorama of the Americas. In *2016 International Symposium on Computers in Education (SIIE)*, pages 1–6. IEEE. <https://doi.org/10.1109/SIIE.2016.7751839>.
- [42] Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., and Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education, WiPSCE '17*, page 65–72, New York, NY, USA. Association for Computing Machinery. <https://doi.org/10.1145/3137065.3137069>.
- [43] Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, volume 1, page 25. https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf.
- [44] British Columbia Ministry of Education (2016). Building student success: Bc's new curriculum. <https://curriculum.gov.bc.ca/curriculum/science>.

- [45] Bruce, R. and Threlfall, J. (2004). One, two, three and counting. *Educational Studies in Mathematics*, 55(1):3–26. <https://doi.org/10.1023/B:EDUC.0000017676.79430.dc>.
- [46] Bybee, R. W. (2014). Ngss and the next generation of science teachers. *Journal of Science Teacher Education*, 25(2):211–221. <https://doi.org/10.1007/s10972-014-9381-4>.
- [47] Campbell-Barr, V., Lavelle, M., and Wickett, K. (2012). Exploring alternative approaches to child outcome assessments in children's centres. *Early Child Development and Care*, 182(7):859–874. <https://doi.org/10.1080/03004430.2011.590937>.
- [48] CARROLL, J. M., SINGLEY, M. K., and ROSSON, M. B. (1992). Integrating theory development with design evaluation. *Behaviour & Information Technology*, 11(5):247–255. <https://doi.org/10.1080/01449299208924345>.
- [49] Caspersen, M. E. and Nowack, P. (2013). Computational thinking and practice: A generic approach to computing in danish high schools. In *Proceedings of the Fifteenth Australasian Computing Education Conference-Volume 136*, pages 137–143. <https://www.cs.au.dk/~mec/publications/conference/41--ace2013.pdf>.
- [50] Castoldi, M. (2009). *Valutare le competenze*. Carocci.
- [51] Chambers, J. M. and Hastie, T. J. (1992). Statistical models. In *Statistical models in S*, pages 13–44. Routledge.
- [52] Chavira, M. and Darwiche, A. (2007). Compiling bayesian networks using variable elimination. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07*, page 2443–2449, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=03f3197f5f2c0415f972111173ed452a39d436d2>.
- [53] Chevalier, M., Giang, C., Piatti, A., and Mondada, F. (2020). Fostering computational thinking through educational robotics: a model for creative computational problem solving (CCPS). *International Journal of STEM Education*, 39. <https://doi.org/10.1186/s40594-020-00238-z>.
- [54] Chiu, M. M. and Mok, M. M. C. (2020). Technology-enhanced learning and assessment. *Educational Psychology*, 40(9):1053–1055. <https://doi.org/10.1080/01443410.2020.1810909>.

- [55] CIIP (2024). Plan d'études romand (per). <https://per.ciip.ch/api/files/270>.
- [56] Claus, F., Rosales, H. G., Petrick, R., Hain, H.-U., and Hoffmann, R. (2013). A survey about databases of children's speech. In *Interspeech 2013*. ISCA. <https://doi.org/10.21437/interspeech.2013-561>.
- [57] Cochran, W. G. (1954). Some methods for strengthening the common χ^2 tests. *Biometrics*, 10(4):417. <http://doi.org/10.2307/3001616>.
- [58] Computer Education Unit (2016). O-level computing syllabus. Sciences Branch, Curriculum Planning and Development Division, Ministry of Education. <https://www.moe.gov.sg/-/media/files/secondary/syllabuses/science/2021-o-level-computing-teaching-and-learning-syllabus.pdf>.
- [59] Corbett, A. T. and Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modelling and User-Adapted Interaction*, 4(4):253–278. <https://doi.org/10.1007/bf01099821>.
- [60] Corecco, S., Adorni, G., and Gambardella, L. M. (2023). Proximal Policy Optimization-Based Reinforcement Learning and Hybrid Approaches to Explore the Cross Array Task Optimal Solution. *Machine Learning and Knowledge Extraction*, 5(4):1660–1679. <http://doi.org/10.3390/make5040082>.
- [61] Cortina, T. J. (2015). Reaching a broader population of students through “unplugged” activities. *Communications of the ACM*, 58(3):25–27. <https://doi.org/10.1145/2723671>.
- [62] Coursaris, C. K. and Kim, D. J. (2011). A meta-analytical review of empirical mobile usability studies. *J. Usability Studies*, 6(3):117–171. <http://uxpajournal.org/a-meta-analytical-review-of-empirical-mobile-usability-studies/>.
- [63] Cox, D. and Hinkley, D. (1979). *Theoretical Statistics*. Chapman and Hall/CRC. <http://doi.org/10.1201/b14832>.
- [64] Crevits, H. (2018). Stem framework for flemish schools-principles and objectives. <https://data-onderwijs.vlaanderen.be/documenten/bestanden/STEM-kader-Engels.pdf>.
- [65] Csernoch, M., Biró, P., Máth, J., and Abari, K. (2015). Testing algorithmic skills in traditional and non-traditional programming environments. *Informatics in Education*, 14(2):175–197. <https://doi.org/10.15388/infedu.2015.11>.

- [66] CSforALL (2018). CS for ALL Students. <https://www.csforall.org/funders/>.
- [67] Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., and Woollard, J. (2015). Computational thinking-a guide for teachers. Technical report, Computing at School. <https://eprints.soton.ac.uk/424545/>.
- [68] Cui, Z. and Ng, O.-L. (2021). The interplay between mathematical and computational thinking in primary school students' mathematical problem-solving within a programming environment. *Journal of Educational Computing Research*, 59(5):988–1012. <https://doi.org/10.1177/0735633120979930>.
- [69] Cyprus Digital Skills and Jobs Coalition (2023). Digital skills - cyprus national action plan 2021-2025. <https://digitalcoalition.gov.cy/strategy/digital-skills-cyprus-national-action-plan-2021-2025/>.
- [70] Czech Ministry of Education, Youth and Sports (2020). Strategy for education policy of the czech republic up to 2030+. https://msmt.gov.cz/uploads/br ozura_S2030_en_fin_online.pdf.
- [71] Dagiene, V. and Stupuriene, G. (2016). Informatics concepts and computational thinking in k-12 education: A lithuanian perspective. *Journal of Information Processing*, 24(4):732–739. <https://doi.org/10.2197/ipsjjip.24.732>.
- [72] Dagienė, V. and Sentance, S. (2016). It's Computational Thinking! Bebras Tasks in the Curriculum. In *International conference on informatics in schools: Situation, evolution, and perspectives*, pages 28–39. Springer, Springer, Cham. https://doi.org/10.1007/978-3-319-46747-4_3.
- [73] Dawson, P. (2017). Assessment rubrics: towards clearer and more replicable design, research and practice. *Assessment & Evaluation in Higher Education*, 42(3):347–360. <https://doi.org/10.1080/02602938.2015.1111294>.
- [74] Dehaene, S. (2011). *The Number Sense: How the Mind Creates Mathematics, Revised and Updated Edition*. Oxford University Press, USA.
- [75] Del Olmo-Muñoz, J., Cózar-Gutiérrez, R., and González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 150:103832. <https://doi.org/10.1016/j.compedu.2020.103832>.

- [76] Delal, H. and Oner, D. (2020). Developing middle school students' computational thinking skills using unplugged computing activities. *Informatics in Education*, 19(1):1–13. <https://doi.org/10.15388/infedu.2020.01>.
- [77] Denning, P. J. and Tedre, M. (2019). *Computational thinking*. Mit Press.
- [78] Denning, P. J. and Tedre, M. (2021). Computational thinking: A disciplinary perspective. *Informatics in Education*. <https://doi.org/10.15388/infedu.2021.21>.
- [79] des Kantons Bern, E. (2016). Lehrplan 21. <https://www.lehrplan21.ch/>.
- [80] Desmarais, M. C. and Baker, R. S. J. d. (2011). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38. <https://doi.org/10.1007/s11257-011-9106-8>.
- [81] Deutschschweizer Erziehungsdirektoren-Konferenz (D-EDK) (2016). Lehrplan 21. <https://v-fe.lehrplan.ch/>.
- [82] Dick, W., Carey, L., and Carey, J. O. (2005). *The systematic design of instruction*. Citeseer. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=671f411d07f151f589184da7d3dbad1d4630d63a>.
- [83] Dietz, G., Landay, J. A., and Gweon, H. (2019). Building blocks of computational thinking: Young children's developing capacities for problem decomposition. In *CogSci*, pages 1647–1653. https://hci.stanford.edu/research/storycoder/pdfs/building_blocks.pdf.
- [84] Digital Skills and Jobs Coalition Luxembourg (2022). Digital skills in education. <https://digitalskills.lu/digital-skills-in-education/>.
- [85] Digital Skills and Jobs Platform (2022). Romania - strategic initiative for digitization of education smart-edu 2021-2027. <https://digital-skills-jobs.europa.eu/en/actions/national-initiatives/national-strategies/romania-strategic-initiative-digitization>.
- [86] Druin, A. (1999). Cooperative inquiry: developing new technologies for children with children. In *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99*, pages 592–599. ACM Press. <https://doi.org/10.1145/302979.303166>.

- [87] Druin, A. (2002). The role of children in the design of new technology. *Behaviour and information technology*, 21(1):1–25. <https://citeseerx.ist.psu.edu/viewdoc/download?amp=&doi=10.1.1.134.4492&rep=rep1&type=pdf>.
- [88] Druin, A., Hammer, J., Kruskal, A., Lal, A., Schwenn, T. P., Sumida, L., Wagner, R., Alborzi, H., Montemayor, J., and Sherman, L. (2000). How do adults and children work together to design new technology? *ACM SIGCHI Bulletin*, 32(2):7–8. <https://doi.org/10.1145/360405.360411>.
- [89] El-Hamamsy, L., Bruno, B., Audrin, C., Chevalier, M., Avry, S., Zufferey, J. D., and Mondada, F. (2023). How are primary school computer science curricular reforms contributing to equity? impact on student learning, perception of the discipline, and gender gaps. *International Journal of STEM Education*, 10(1). <https://doi.org/10.1186/s40594-023-00438-3>.
- [90] El-Hamamsy, L., Zapata-Cáceres, M., Barroso, E. M., Mondada, F., Zufferey, J. D., and Bruno, B. (2022). The competent computational thinking test: Development and validation of an unplugged computational thinking test for upper primary school. *Journal of Educational Computing Research*, 60(7):1818–1866. <http://doi.org/10.1177/07356331221081753>.
- [91] Embretson, S. E. and Reise, S. P. (2013). *Item response theory*. Psychology Press.
- [92] European Schoolnet (2018). Country report on ict in education. http://www.eun.org/documents/411753/839549/Country+Report+Norway+2018_v2.pdf/e8c32816-d56e-4080-8154-d2f6ca6f9961.
- [93] Ezeamuzie, N. O. and Leung, J. W. (2021). Computational thinking through an empirical lens: a systematic review of literature. *Journal of Educational Computing Research*, 60(2):481–511. <https://doi.org/10.1177/07356331211033158>.
- [94] Fainberg, J., Bell, P., Lincoln, M., and Renals, S. (2016). Improving children’s speech recognition through out-of-domain data augmentation. In *Interspeech 2016*, page 1598–1602. ISCA. <https://doi.org/10.21437/interspeech.2016-1348>.
- [95] Finnish National Board of Education (FNBE) (2016). National core curriculum for basic education 2014. <https://www.oph.fi/en/education-and-qualifications/national-core-curriculum-primary-and-lower-secondary-basic-education>.

- [96] Fisk, P. R. and Weisberg, S. (1982). Applied linear regression. *Journal of the Royal Statistical Society. Series A (General)*, 145(1):146. <http://doi.org/10.2307/2981445>.
- [97] Fraillon, J., Ainley, J., Schulz, W., Friedman, T., and Duckworth, D. (2020). *Preparing for Life in a Digital World: IEA International Computer and Information Literacy Study 2018 International Report*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-38781-5>.
- [98] French Government (2015). Socle commun de connaissances, de compétences et de culture. <https://www.education.gouv.fr/bo/15/Hebdo17/MENE1506516D.htm>.
- [99] Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. In *Informatics Education – The Bridge between Using and Understanding Computers*, pages 159–168. Springer Berlin Heidelberg. https://doi.org/10.1007/11915355_15.
- [100] Gelman, R. and Gallistel, C. R. (1986). *The Child’s Understanding of Number*. Harvard University Press.
- [101] Georgiou, K. and Angeli, C. (2021). *Developing Computational Thinking in Early Childhood Education: A Focus on Algorithmic Thinking and the Role of Cognitive Differences and Scaffolding*, pages 33–49. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-65657-7_3.
- [102] Giannakos, M., Markopoulos, P., Hourcade, J. P., and Antle, A. N. (2022). ‘lots done, more to do’: The current state of interaction design and children research and future directions. *International Journal of Child-Computer Interaction*, 33:100469. <https://doi.org/10.1016/j.ijcci.2022.100469>.
- [103] Google and community (2017). Flutter framework. <https://flutter.dev/>.
- [104] Google, MIT (2012). Blockly. <https://developers.google.com/blockly/>.
- [105] Gould, J. D. and Lewis, C. (1985). Designing for usability: key principles and what designers think. *Communications of the ACM*, 28(3):300–311. <https://doi.org/10.1145/3166.3170>.
- [106] Gouws, L. A., Bradshaw, K., and Wentworth, P. (2013). Computational thinking in educational activities: an evaluation of the educational game light-bot. In

Proceedings of the 18th ACM conference on Innovation and technology in computer science education, ITiCSE '13. ACM.

- [107] Government of Ireland (2018). Computer science: Curriculum specification. <https://curriculumonline.ie/getmedia/d73af6e3-b4e5-4edb-a514-6383e2306a4b/16626-NCCASpecification-for-Leaving-Certificate-CS-WEB-v4.pdf>.
- [108] Greenbaum, J. and Kyng, M. (2020). *Design at work: Cooperative design of computer systems*. CRC Press.
- [109] Greig, A. D., Taylor, J., and MacKay, T. (2007). *Doing research with children*. Sage.
- [110] Greig, A. D., Taylor, J., and MacKay, T. (2012). *Doing research with children: A practical guide*. Sage.
- [111] Grgurina, N., Barendsen, E., Zwaneveld, B., van de Grift, W., and Stoker, I. (2013). Computational thinking skills in dutch secondary education. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education*, volume 12 of *WiPSCE '13*, page 31–32. ACM. <https://doi.org/10.1145/2532748.2532768>.
- [112] Grover, S. (2017). Assessing algorithmic and computational thinking in k-12: Lessons from a middle school classroom. *Emerging research, practice, and policy on computational thinking*, pages 269–288. https://doi.org/10.1007/978-3-319-52691-1_17.
- [113] Grover, S., Cooper, S., and Pea, R. (2014). Assessing Computational Learning in K-12. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, page 57–62, New York, NY, USA. Association for Computing Machinery. <https://doi.org/10.1145/2591708.2591713>.
- [114] Grover, S. and Pea, R. (2013a). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1):38–43. <https://doi.org/10.3102/0013189X12463051>.
- [115] Grover, S. and Pea, R. (2017). Computational Thinking: A Competency Whose Time Has Come. *Computer science education: Perspectives on teaching and learning in school*, 19:1257–1258. <https://doi.org/10.5040/9781350057142.ch-003>.

- [116] Grover, S., Pea, R., and Cooper, S. (2015a). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2):199–237. <https://doi.org/10.1080/08993408.2015.1033142>.
- [117] Guo, L., Wang, D., Gu, F., Li, Y., Wang, Y., and Zhou, R. (2021). Evolution and trends in intelligent tutoring systems research: a multidisciplinary and scientometric view. *Asia Pacific Education Review*, 22(3):441–461. <https://doi.org/10.1007/s12564-021-09697-7>.
- [118] Guran, A.-M., Cojocar, G.-S., and Turian, A. (2020). Towards preschoolers' automatic satisfaction assessment. an experience report. In *2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE. <https://doi.org/10.1109/saci49304.2020.9118824>.
- [119] Gurunath Shivakumar, P. and Georgiou, P. (2020). Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations. *Computer Speech & Language*, 63:101077. <https://doi.org/10.1016/j.csl.2020.101077>.
- [120] Gurunath Shivakumar, P. and Narayanan, S. (2022). End-to-end neural systems for automatic children speech recognition: An empirical study. *Computer Speech & Language*, 72:101289. <https://doi.org/10.1016/j.csl.2021.101289>.
- [121] Hanington, B. and Martin, B. (2019). *Universal methods of design expanded and revised: 125 Ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport publishers.
- [122] Hanna, L., Ridsen, K., and Alexander, K. (1997). Guidelines for usability testing with children. *Interactions*, 4(5):9–14. <https://doi.org/10.1145/264044.264045>.
- [123] Hartson, R. and Pyla, P. S. (2018). *The UX book: Agile UX design for a quality user experience*. Morgan Kaufmann.
- [124] Hastie, T., Friedman, J., and Tibshirani, R. (2001). *The Elements of Statistical Learning*. Springer New York. <http://doi.org/10.1007/978-0-387-21606-5>.
- [125] Heersmink, R. (2013). A taxonomy of cognitive artifacts: Function, information, and categories. *Review of Philosophy and Psychology*, 4(3):465–481. <https://doi.org/10.1007/s13164-013-0148-1>.

- [126] Heintz, F., Mannila, L., Nygård, K., Parnes, P., and Regnell, B. (2015). *Computing at School in Sweden – Experiences from Introducing Computer Science within Existing Subjects*, page 118–130. Springer International Publishing. https://doi.org/10.1007/978-3-319-25396-1_11.
- [127] Hermans, F. and Aivaloglou, E. (2017). To Scratch or not to Scratch?: A controlled experiment comparing plugged first and unplugged first programming lessons. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, pages 49–56. ACM. <https://doi.org/10.1145/3137065.3137072>.
- [128] High-Level Group on International Education to the Tánaiste and Minister for Education and Skills (2010). Investing in global relationships: Ireland’s international education strategy 2010-15. <https://assets.gov.ie/24400/a46c03d90bf24e54a83776a383baa9d4.pdf>.
- [129] Hinckle, M., Rachmatullah, A., Mott, B., Boyer, K. E., Lester, J., and Wiebe, E. (2020). The relationship of gender, experiential, and psychological factors to achievement in computer science. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE ’20*. ACM. <https://doi.org/10.1145/3341525.3387403>.
- [130] Hoe, S. L. (2016). Defining a smart nation: the case of singapore. *Journal of Information, Communication and Ethics in Society*, 14(4):323–333. <https://doi.org/10.1108/jices-02-2016-0005>.
- [131] Holtzblatt, K. and Beyer, H. (1997). *Contextual design: defining customer-centered systems*. Elsevier.
- [132] Hooshyar, D., Ahmad, R. B., Yousefi, M., Fathi, M., Horng, S.-J., and Lim, H. (2016a). SITS: A solution-based intelligent tutoring system for students’ acquisition of problem-solving skills in computer programming. *Innovations in Education and Teaching International*, 55(3):325–335. <https://doi.org/10.1080/14703297.2016.1189346>.
- [133] Hooshyar, D., Ahmad, R. B., Yousefi, M., Fathi, M., Horng, S.-J., and Lim, H. (2016b). Sits: A solution-based intelligent tutoring system for students’ acquisition of problem-solving skills in computer programming. *Innovations in Education and Teaching International*, 55(3):325–335. <http://doi.org/10.1080/14703297.2016.1189346>.

- [134] Hörmann, C. (2023). *The Journey of Digital Education in Austria - From Non-Existent to Mandatory in Five Years*. PhD thesis, Johannes Kepler University Linz. <https://doi.org/10.13140/RG.2.2.10740.73608>.
- [135] Hourcade, J. P. (2007). Interaction design and children. *Foundations and Trends® in Human-Computer Interaction*, 1(4):277–392. <https://doi.org/10.1561/11000000006>.
- [136] House, W. (2016). President obama announces computer science for all initiative. <https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>.
- [137] Hox, J., Moerbeek, M., and Van de Schoot, R. (2017). *Multilevel analysis: Techniques and applications*. Routledge.
- [138] Hsu, T.-C., Chang, S.-C., and Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126:296–310. <https://doi.org/10.1016/j.compedu.2018.07.004>.
- [139] Huber, D., Cabañas, R., Antonucci, A., and Zaffalon, M. (2020). Crema: A java library for credal network inference. In Jaeger, M. and Nielsen, T. D., editors, *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, pages 613–616, Skørping, Denmark. PMLR. <https://proceedings.mlr.press/v138/huber20a.html>.
- [140] Hungary Government (2016). The digital education strategy of hungary. <https://2015-2019.kormany.hu/download/0/4b/21000/The%20Digital%20Education%20Strategy%20of%20Hungary.pdf>.
- [141] Iivari, N. and Kinnula, M. (2018). Empowering children through design and making. In *Proceedings of the 15th Participatory Design Conference: Full Papers - Volume 1*. ACM. <https://doi.org/10.1145/3210586.3210600>.
- [142] Iversen, O. S. and Dindler, C. (2013). A utopian agenda in child–computer interaction. *International Journal of Child-Computer Interaction*, 1(1):24–29. <https://doi.org/10.1016/j.ijcci.2012.08.002>.
- [143] Iversen, O. S. and Smith, R. C. (2012). Scandinavian participatory design: dialogic curation with teenagers. In *Proceedings of the 11th International Conference on Interaction Design and Children*. ACM. <https://doi.org/10.1145/2307096.2307109>.

- [144] Iversen, O. S., Smith, R. C., and Dindler, C. (2017). Child as protagonist: Expanding the role of children in participatory design. In *Proceedings of the 2017 Conference on Interaction Design and Children*, pages 27–37. ACM. <https://doi.org/10.1145/3078072.3079725>.
- [145] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer New York. <http://doi.org/10.1007/978-1-4614-7138-7>.
- [146] Jiang, S. and Wong, G. K. W. (2022). Exploring age and gender differences of computational thinkers in primary school: A developmental perspective. *Journal of Computer Assisted Learning*, 38(1):60–75. <https://doi.org/10.1111/jcal.12591>.
- [147] Jocz, J., Peterson, K. A., and Pfeif, D. (2023). Motivating youth to learn STEM through a gender inclusive digital forensic science program. *Smart Learning Environments*, 10(1). <https://doi.org/10.1186/s40561-022-00213-x>.
- [148] Jonsson, A. and Svingby, G. (2007). The use of scoring rubrics: Reliability, validity and educational consequences. *Educational Research Review*, 2(2):130–144. <https://doi.org/10.1016/j.edurev.2007.05.002>.
- [149] Kafai, Y. B., Ching, C. C., and Marshall, S. (1997). Children as designers of educational multimedia software. *Computers & Education*, 29(2-3):117–126. [http://doi.org/10.1016/s0360-1315\(97\)00036-5](http://doi.org/10.1016/s0360-1315(97)00036-5).
- [150] Kafai, Y. B., Proctor, C., and Lui, D. (2020). From Theory Bias to Theory Dialogue: Embracing Cognitive, Situated, and Critical Framings of Computational Thinking in K-12 CS Education. *ACM Inroads*, 11(1):44–53. <https://doi.org/10.1145/3381887>.
- [151] Kalas, I. (2015). Programming at pre-primary and primary levels: the pipeline can start that early. *KEYCIT 2014*, page 29.
- [152] Kalelioglu, F., Gulbahar, Y., and Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing*, 4(3):583–596. https://www.bjmc.lu.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/Contents/4_3_15_Kalelioglu.pdf.
- [153] Kali, Y. and Fuhrmann, T. R. (2011). Teaching to design educational technologies. *International Journal of Learning Technology*, 6(1):4. <https://doi.org/10.1504/ijlt.2011.040147>.

- [154] Kanaki, K. and Kalogiannakis, M. (2022). Assessing algorithmic thinking skills in relation to age in early childhood stem education. *Education Sciences*, 12(6). <https://doi.org/10.3390/educsci12060380>.
- [155] Kathania, H. K., Ahmad, W., Shahnawazuddin, S., and Samaddar, A. B. (2018a). Explicit pitch mapping for improved children's speech recognition. *Circuits, Systems, and Signal Processing*, 37(5):2021–2044. <https://doi.org/10.1007/s00034-017-0652-0>.
- [156] Kathania, H. K., Shahnawazuddin, S., Ahmad, W., Adiga, N., Jana, S. K., and Samaddar, A. B. (2018b). Improving children's speech recognition through time scale modification based speaking rate adaptation. In *2018 International Conference on Signal Processing and Communications (SPCOM)*, pages 257–261. <https://doi.org/10.1109/SPCOM.2018.8724465>.
- [157] Keith, P. K., Sullivan, F. R., and Pham, D. (2019). Roles, collaboration, and the development of computational thinking in a robotics learning environment. In *Computational Thinking Education*, pages 223–245. Springer Singapore. https://doi.org/10.1007/978-981-13-6528-7_13.
- [158] Kinnula, M. and Iivari, N. (2021). Manifesto for children's genuine participation in digital technology design and making. *International Journal of Child-Computer Interaction*, 28:100244. <https://doi.org/10.1016/j.ijcci.2020.100244>.
- [159] Kinnula, M., Iivari, N., Molin-Juustila, T., Keskitalo, E., and Leinonen, T. (2017). Cooperation, combat, or competence building: what do we mean when we are 'empowering children'in and through digital technology design? In *Proceedings of International Conference on Information Systems (ICIS 2017). 10-12 December 2017, Seoul, Korea*. International Conference on Informations Systems. <https://aisel.aisnet.org/icis2017/TransformingSociety/Presentations/15>.
- [160] Klahr, D. and Robinson, M. (1981). Formal assessment of problem-solving and planning processes in preschool children. *Cognitive Psychology*, 13(1):113–148. [https://doi.org/10.1016/0010-0285\(81\)90006-2](https://doi.org/10.1016/0010-0285(81)90006-2).
- [161] Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.
- [162] Kong, S.-C., Abelson, H., and Lai, M. C. (2019). *Introduction To Computational Thinking Education*, pages 1–10. Springer Singapore. https://doi.org/10.1007/978-981-13-6528-7_1.

- [163] Kong, S.-C., Chiu, M. M., and Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education*, 127:178–189. <https://doi.org/10.1016/j.compedu.2018.08.026>.
- [164] Kong, S.-C. and Lai, M. (2022). Validating a computational thinking concepts test for primary education using item response theory: An analysis of students' responses. *Computers & Education*, 187:104562. <https://doi.org/10.1016/j.compedu.2022.104562>.
- [165] Korkmaz, Ö. and Bai, X. (2019). Adapting Computational Thinking Scale (CTS) for Chinese High School Students and Their Thinking Scale Skills Level. *Participatory Educational Research*, 6(1):10–26. <https://doi.org/10.17275/per.19.2.6.1>.
- [166] Korkmaz, Ö., Çakır, R., and Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (cts). *Computers in Human Behavior*, 72:558–569. <https://doi.org/10.1016/j.chb.2017.01.005>.
- [167] Kralj, L. (2016). New informatics curriculum — croatian tradition with world trends. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 760–763. <https://doi.org/10.1109/MIPRO.2016.7522242>.
- [168] Kujala, S. (2003). User involvement: A review of the benefits and challenges. *Behaviour & Information Technology*, 22(1):1–16. <https://doi.org/10.1080/01449290301782>.
- [169] Kules, B. (2016). Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes. *Proceedings of the Association for Information Science and Technology*, 53(1):1–6. <https://doi.org/10.1002/pra2.2016.14505301092>.
- [170] Kuznetsova, A., Brockhoff, P. B., and Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, 82(13). <https://doi.org/10.18637/jss.v082.i13>.
- [171] Lafuente Martínez, M., Lévêque, O., Benítez, I., Hardebolle, C., and Zufferey, J. D. (2022). Assessing computational thinking: Development and validation of the algorithmic thinking test for adults. *Journal of Educational Computing Research*, 60(6):1436–1463. <https://doi.org/10.1177/07356331211057819>.

- [172] Lee, H. L. (2014). Why smart nation: our vision. <https://www.smartnation.gov.sg/media-hub/speeches/smart-nation-launch/>.
- [173] Lehnert, F. K., Niess, J., Lallemand, C., Markopoulos, P., Fischbach, A., and Koenig, V. (2022). Child-computer interaction: From a systematic review towards an integrated understanding of interaction design methods for children. *International Journal of Child-Computer Interaction*, 32:100398. <https://doi.org/10.1016/j.ijcci.2021.100398>.
- [174] Lenth, R. V. (2023). *emmeans: Estimated Marginal Means, aka Least-Squares Means*. R package version 1.9.0; <https://CRAN.R-project.org/package=emmeans>.
- [175] Lodi, M. and Martini, S. (2021). Computational thinking, between papert and wing. *Science & Education*, 30(4):883–908. <https://doi.org/10.1007/s11191-021-00202-5>.
- [176] Lu, J. J. and Fletcher, G. H. L. (2009). Thinking About Computational Thinking. In *Proceedings of the 40th ACM technical symposium on Computer science education, SIGCSE '09*, pages 260–264. <https://doi.org/10.1145/1508865.1508959>.
- [177] Lui, A. L. C., Not, C., and Wong, G. K. W. (2023). Theory-based learning design with immersive virtual reality in science education: a systematic review. *Journal of Science Education and Technology*, 32(3):390–432. <https://doi.org/10.1007/s10956-023-10035-2>.
- [178] Mahone, E. M. and Schneider, H. E. (2012). Assessment of attention in preschoolers. *Neuropsychology Review*, 22(4):361–383. <https://doi.org/10.1007/s11065-012-9217-y>.
- [179] Makransky, G. and Petersen, G. B. (2021). The cognitive affective model of immersive learning (camil): A theoretical research-based model of learning in immersive virtual reality. *Educational Psychology Review*, 33(3):937–958. <https://doi.org/10.1007/s10648-020-09586-2>.
- [180] Maloney, J., Burd, L., Kafai, Y. B., Rusk, N., Silverman, B., and Resnick, M. (2004). Scratch: a sneak preview [education]. In *Proceedings. Second International Conference on Creating, Connecting and Collaborating through Computing, 2004.*, pages 104–109. <https://doi.org/10.1109/C5.2004.1314376>.
- [181] Mangili, F., Adorni, G., Piatti, A., Bonesana, C., and Antonucci, A. (2022). Modelling Assessment Rubrics through Bayesian Networks: a Pragmatic Approach. In

- 2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE. <https://doi.org/10.23919/softcom55329.2022.9911432>.
- [182] Markopoulos, P. and Bekker, M. (2003). Interaction design and children. *Interacting with Computers*, 15(2):141–149. [https://doi.org/10.1016/s0953-5438\(03\)00004-3](https://doi.org/10.1016/s0953-5438(03)00004-3).
- [183] Markopoulos, P., Bekker, M., et al. (2002). How to compare usability testing methods with children participants. In *Interaction Design and Children*, volume 2, pages 28–34. Shaker Publisher.
- [184] Martin, N. and Maes, H. (1979). Multivariate analysis. London, UK: Academic. <https://statisticalsupportandresearch.files.wordpress.com/2017/06/k-v-mardia-j-t-kent-j-m-bibby-multivariate-analysis-probability-and-mathematical-statistics-academic-press-inc-1979.pdf>.
- [185] Mason, S. L. and Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education*, 19(4):790–824. <https://citejournal.org/wp-content/uploads/2019/11/v19i4General1.pdf>.
- [186] Master, A., Meltzoff, A. N., and Cheryan, S. (2021). Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. *Proceedings of the National Academy of Sciences*, 118(48). <https://doi.org/10.1073/pnas.2100030118>.
- [187] Matović, M. (2021). *Digital Competence Programs in the Republic of Serbia*. OSCE. <https://www.osce.org/files/f/documents/7/c/495178.pdf>.
- [188] Mayo, M. J. (2001). Bayesian student modelling and decision-theoretic selection of tutorial actions in intelligent tutoring systems. PhD thesis, University of Canterbury, Computer Science. <https://doi.org/10.26021/1918>.
- [189] McCormick, K. I. and Hall, J. A. (2022). Computational thinking learning experiences, outcomes, and research in preschool settings: a scoping review of literature. *Education and Information Technologies*, pages 1–36. <https://doi.org/10.1007/s10639-021-10765-z>.
- [190] Metin, S. (2020). Activity-based unplugged coding during the preschool period. *International Journal of Technology and Design Education*, pages 1–17. <https://doi.org/10.1007/s10798-020-09616-8>.

- [191] Millán, E., Pérez-de-la Cruz, J. L., and Suárez, E. (2000). *Adaptive Bayesian Networks for Multilevel Student Modelling*, page 534–543. Springer Berlin Heidelberg. http://doi.org/10.1007/3-540-45108-0_57.
- [192] Ministry of Education and Employment (2012). A national curriculum framework for all. <https://curriculum.gov.mt/en/Resources/The-NCF/Documents/NCF.pdf>.
- [193] Ministry of Education and Research of Italy (2015). Piano nazionale scuola digitale. MIUR. https://www.istruzione.it/scuola_digitale/allegati/Materiali/pnsd-layout-30.10-WEB.pdf.
- [194] Ministry of Education New Zealand (2016). Digital technologies fund opens. Education Gazette. <https://gazette.education.govt.nz/articles/digital-technologies-fund-opens/>.
- [195] Moore, D. S. and McCabe, G. P. (1989). *Introduction to the practice of statistics*. Introduction to the practice of statistics. W H Freeman/Times Books/ Henry Holt & Co, New York, NY, US.
- [196] Moreno-León, J., Robles, G., and Román-González, M. (2015). Dr. scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, 46:1–23.
- [197] Mousavinasab, E., Zarifsanaiey, N., Kalhori, S. R. N., Rakhshan, M., Keikha, L., and Saeedi, M. G. (2018). Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1):142–163. <https://doi.org/10.1080/10494820.2018.1558257>.
- [198] Mouza, C., Pan, Y.-C., Yang, H., and Pollock, L. (2020). A multiyear investigation of student computational thinking concepts, practices, and perspectives in an after-school computing program. *Journal of Educational Computing Research*, 58(5):1029–1056. <https://doi.org/10.1177/0735633120905605>.
- [199] Mühling, A., Ruf, A., and Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, WiPSCE '15. ACM. <https://doi.org/10.1145/2818314.2818320>.
- [200] Muller, M. J. and Kuhn, S. (1993). Participatory design. *Communications of the ACM*, 36(6):24–28. <https://doi.org/10.1145/153571.255960>.

- [201] Muller, M. J., Wildman, D. M., and White, E. A. (1993). “equal opportunity” PD using PICTIVE. *Communications of the ACM*, 36(6):64. <https://doi.org/10.1145/153571.214818>.
- [202] Muller, M. J., Wildman, D. M., and White, E. A. (1994). Participatory design through games and other group exercises. In *Conference companion on Human factors in computing systems - CHI '94*. ACM Press. <https://doi.org/10.1145/259963.260530>.
- [203] Muppalla, S. K., Vuppalapati, S., Reddy Pulliahgaru, A., and Sreenivasulu, H. (2023). Effects of excessive screen time on child development: An updated review and strategies for management. *Cureus*. <https://doi.org/10.7759/cureus.40608>.
- [204] Nayebi, F., Desharnais, J.-M., and Abran, A. (2012). The state of the art of mobile application usability evaluation. In *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE. <https://doi.org/10.1109/ccece.2012.6334930>.
- [205] Next Generation Science Standards (2013). Using mathematics and computational thinking. <https://www.nextgenscience.org/practice/using-mathematics-and-computational-thinking>.
- [206] Nielsen, J. (1994b). Usability inspection methods. In *Conference companion on Human factors in computing systems - CHI '94*. ACM Press. <https://doi.org/10.1145/259963.260531>.
- [207] Nielsen, J. (1995). Scenarios in discount usability engineering. In *Scenario-Based Design: Envisioning work and technology in system development*, pages 59–83. John Wiley & Sons, Inc.
- [208] Nikolopoulou, K. and Tsimperidis, I. (2023). Stem education in early primary years: Teachers’ views and confidence. *Journal of Digital Educational Technology*, 3(1):ep2302. <https://doi.org/10.30935/jdet/12971>.
- [209] OECD (2017). Education policy in greece - a preliminary assessment. <https://www.esos.gr/sites/default/files/articles-legacy/education-policy-in-greece-preliminary-assessment-2017.pdf>.
- [210] OECD (2020). *PISA 2018 Results (Volume VI): Are students ready to thrive in an interconnected world?* OECD: Organisation for Economic Co-operation and Development. <https://doi.org/10.1787/d5f68679-en>.

- [211] OECD (2023). *PISA 2022 Mathematics Framework*, chapter PISA 2022 Assessment and Analytical Framework. OECD Publishing. <https://doi.org/10.1787/7ea9ee19-en>.
- [212] Olivier, E., Archambault, I., De Clercq, M., and Galand, B. (2018). Student self-efficacy, classroom engagement, and academic achievement: Comparing three theoretical frameworks. *Journal of Youth and Adolescence*, 48(2):326–340. <https://doi.org/10.1007/s10964-018-0952-0>.
- [213] Olkhova, N. V. (2022). Development of algorithmic thinking in primary school students when studying computer science. *Scientific Bulletin of Mukachevo State University. Series Pedagogy and Psychology*, 8(2):25–32. [https://doi.org/10.52534/msu-pp.8\(2\).2022.25-32](https://doi.org/10.52534/msu-pp.8(2).2022.25-32).
- [214] Oluk, A. and Korkmaz, Ö. (2016). Comparing students' scratch skills with their computational thinking skills in terms of different variables. *International Journal of Modern Education and Computer Science*, 8(11):1–7. <https://doi.org/10.5815/ijmecs.2016.11.01>.
- [215] Oyelere, S. S., Agbo, F. J., and Sanusi, I. T. (2022). Developing a pedagogical evaluation framework for computational thinking supporting technologies and tools. *Frontiers in Education*, 7. <https://doi.org/10.3389/feduc.2022.95739>.
- [216] Pan, Z., Zhang, D., Hu, T., and Pan, Y. (2018). The relationship between psychological suzhi and social anxiety among chinese adolescents: the mediating role of self-esteem and sense of security. *Child and Adolescent Psychiatry and Mental Health*, 12(1). <https://doi.org/10.1186/s13034-018-0255-y>.
- [217] Papert, S. (1980). Mindstorms: children, computers, and powerful ideas. <http://dl.acm.org/doi/book/10.5555/1095592>.
- [218] Papert, S. (2000). What's the big idea? toward a pedagogy of idea power. *IBM Systems Journal*, 39(3.4):720–729. <https://doi.org/10.1147/sj.393.0720>.
- [219] Park, R. K. E. (2016). *Preparing students for South Korea's creative economy: The successes and challenges of educational reform*. Asia Pacific Foundation of Canada. https://www.asiapacific.ca/sites/default/files/filefield/south_korea_education_report_updated.pdf.
- [220] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann. doi: 10.1016/c2009-0-27609-4.

- [221] Perneger, T. V. (1998). What's wrong with bonferroni adjustments. *BMJ*, 316(7139):1236–1238. <http://doi.org/10.1136/bmj.316.7139.1236>.
- [222] Petousi, V. and Sifaki, E. (2020). Contextualising harm in the framework of research misconduct. findings from discourse analysis of scientific publications. *International Journal of Sustainable Development*, 23(3-4):149–174. <https://doi.org/10.1504/IJSD.2020.115206>.
- [223] Piaget, J. (1964). Development and learning. *Journal, of Research in Science Teaching*, 2:176–186.
- [224] Piaget, J., Cook, M., et al. (1952). *The origins of intelligence in children*. W W Norton & Co. <https://doi.org/10.1037/11494-000>.
- [225] Piaget, J. and Mussen, P. (1983). Handbook of child psychology. *History, theory, and methods*.
- [226] Piatti, A. and Adorni, G. (2024). Unplugged Cross Array Task (CAT) Assessment: Supplementary Documentation and Experimental Protocol. Zenodo. <https://doi.org/10.5281/zenodo.12806226>.
- [227] Piatti, A., Adorni, G., El-Hamamsy, L., Negrini, L., Assaf, D., Gambardella, L., and Mondada, F. (2022). The CT-cube: A framework for the design and the assessment of computational thinking activities. *Computers in Human Behavior Reports*, 5:100166. <https://doi.org/10.1016/j.chbr.2021.100166>.
- [228] Pilotti, M., Nazeeruddin, E., Mohammad, N., Daqqa, I., Abdelsalam, H., and Abdullah, M. M. (2022). Is initial performance in a course informative? machine learning algorithms as aids for the early detection of at-risk students. *Electronics*, 11(13):2057. <https://doi.org/10.3390/electronics11132057>.
- [229] Plante, I., de la Sablonnière, R., Aronson, J. M., and Théorêt, M. (2013). Gender stereotype endorsement and achievement-related outcomes: The role of competence beliefs and task values. *Contemporary Educational Psychology*, 38(3):225–235. <https://doi.org/10.1016/j.cedpsych.2013.03.004>.
- [230] Polat, E., Hopcan, S., Kucuk, S., and Sisman, B. (2021). A comprehensive assessment of secondary school students' computational thinking skills. *British Journal of Educational Technology*, 52(5):1965–1980. <https://doi.org/10.1111/bjet.13092>.

- [231] Ponti, M. (2023). Screen time and preschool children: Promoting health and development in a digital world. *Paediatrics & Child Health*, 28(3):184–192. <https://doi.org/10.1093/pch/pxac125>.
- [232] Potamianos, A. and Narayanan, S. (2003). Robust recognition of children’s speech. *IEEE Transactions on Speech and Audio Processing*, 11(6):603–616. <https://doi.org/10.1109/tsa.2003.818026>.
- [233] Poulakis, E. and Politis, P. (2021). *Computational Thinking Assessment: Literature Review*, pages 111–128. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-64363-8_7.
- [234] Qian, Y. and Lehman, J. D. (2018). Using technology to support teaching computer science: a study with middle school students. *Eurasia Journal of Mathematics Science and Technology Education*, 14(12). <https://doi.org/10.29333/ejmste/94227>.
- [235] R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- [236] Rachmatullah, A., Vandenberg, J., and Wiebe, E. (2022). Toward more generalizable cs and ct instruments: Examining the interaction of country and gender at the middle grades level. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, ITiCSE ’22, page 179–185, New York, NY, USA. Association for Computing Machinery. <https://doi.org/10.1145/3502718.3524790>.
- [237] Rapaport, W. J. (2015). Philosophy of computer science. *Teaching philosophy*. Current draft in progress at <https://cse.buffalo.edu/~rapaport/Papers/phics.pdf>.
- [238] Raudenbush, S. W. and Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods*, volume 1. sage.
- [239] Read, J. C. and MacFarlane, S. (2006). Using the fun toolkit and other survey methods to gather opinions in child computer interaction. In *Proceedings of the 2006 conference on Interaction design and children*. ACM. <https://doi.org/10.1145/1139073.1139096>.
- [240] Read, J. C., MacFarlane, S., and Gregory, P. (2004). Requirements for the design of a handwriting recognition based writing interface for children. In *Proceedings of*

- the 2004 conference on Interaction design and children: building a community*. ACM. <https://doi.org/10.1145/1017833.1017844>.
- [241] Relkin, E. and Bers, M. (2021). TechCheck-K: A Measure of Computational Thinking for Kindergarten Children. In *2021 IEEE Global Engineering Education Conference (EDUCON)*, pages 1696–1702. <https://doi.org/10.1109/EDUCON46332.2021.9453926>.
- [242] Relkin, E., de Ruiter, L., and Bers, M. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, 169:104222. <https://doi.org/10.1016/j.compedu.2021.104222>.
- [243] Relkin, E., de Ruiter, L., and Bers, M. U. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology*, 29(4):482–498. <https://doi.org/10.1007/s10956-020-09831-x>.
- [244] Relkin, E. and Umaschi Bers, M. (2019). *Designing an Assessment of Computational Thinking Abilities for Young Children*, page 83–98. Routledge. <https://doi.org/10.4324/9780429453755-5>.
- [245] Repubblica e Cantone Ticino, Dipartimento dell’educazione, della cultura e dello sport (2015). Piano di studio della scuola dell’obbligo ticinese. <https://scuolalab.edu.ti.ch/temieprogetti/pds/PublishingImages/Piano%20di%20studio%20della%20scuola%20dell%27obbligo%20ticinese%20-%20VERSIONE%202015.pdf>.
- [246] Repubblica e Cantone Ticino, Dipartimento dell’educazione, della cultura e dello sport (2022). Piano di studio della scuola dell’obbligo ticinese. <https://scuolalab.edu.ti.ch/temieprogetti/pds/Documents/Perfezionato/Piano%20di%20studio%20perfezionato.pdf>.
- [247] Republic of Slovenia: Ministry of Education (2020). Digital education action plan (andi) 2021-2027. <https://www.gov.si/assets/ministrstva/MVI/SDIG/SI-Digital-Education-Action-Plan-EN-web.pdf>.
- [248] Rodriguez, B., Rader, C., and Camp, T. (2016). Using student performance to assess CS unplugged activities in a classroom environment. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 95–100. ACM. <https://doi.org/10.1145/2899415.2899465>.

- [249] Rodríguez-Barrios, E. U., Melendez-Armenta, R. A., García-Aburto, S. G., Lavoignet-Ruiz, M., Sandoval-Herazo, L. C., Molina-Navarro, A., and Morales-Rosales, L. A. (2021). Bayesian approach to analyze reading comprehension: A case study in elementary school children in Mexico. *Sustainability*, 13(8):4285. <https://doi.org/10.3390/su13084285>.
- [250] Rodríguez-Martínez, J. A., González-Calero, J. A., and Sáez-López, J. M. (2019). Computational thinking and mathematics using scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, 28(3):316–327. <https://doi.org/10.1080/10494820.2019.1612448>.
- [251] Román-González, M. (2015). Computational thinking test: Design guidelines and content validation. In *EDULEARN15 Proceedings*, pages 2436–2444. IATED. https://www.researchgate.net/publication/290391277_COMPUTATIONAL_THINKING_TEST_DESIGN_GUIDELINES_AND_CONTENT_VALIDATION.
- [252] Román-González, M., Moreno-León, J., and Robles, G. (2017). Complementary Tools for Computational Thinking Assessment. In *Proceedings of International Conference on Computational Thinking Education (CTE 2017)*, S. C Kong, J Sheldon, and K. Y Li (Eds.). The Education University of Hong Kong, pages 154–159. <https://www.eduhk.hk/cte2017/doc/CTE2017%20Proceedings.pdf>.
- [253] Román-González, M., Moreno-León, J., and Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. *Computational thinking education*, pages 79–98. https://doi.org/10.1007/978-981-13-6528-7_6.
- [254] Romero, M., Lepage, A., and Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, 14(1):1–15. <https://doi.org/10.1186/s41239-017-0080-z>.
- [255] Román-González, M. and Pérez-González, J.-C. (2024). *Computational Thinking Assessment: A Developmental Approach*, page 121–142. The MIT Press. <http://dx.doi.org/10.7551/mitpress/14041.003.0009>.
- [256] Román-González, M., Pérez-González, J.-C., and Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in Human Behavior*, 72:678–691. <https://doi.org/10.1016/j.chb.2016.08.047>.

- [257] Román-González, M., Pérez-González, J.-C., Moreno-León, J., and Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction*, 18:47–58. <https://doi.org/10.1016/j.ijcci.2018.06.004>.
- [258] Roth, W.-M. and Jornet, A. (2013). Situated cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 4(5):463–478. <https://doi.org/10.1002/wcs.1242>.
- [259] Sabitzer, B., Antonitsch, P. K., and Pasterk, S. (2014). Informatics concepts for primary education: preparing children for computational thinking. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, WiPSCE '14, pages 108–111. <https://doi.org/10.1145/2670757.2670778>.
- [260] Sarama, J. and Clements, D. H. (2009). *Early childhood mathematics education research: Learning trajectories for young children*. Routledge.
- [261] Sarnecka, B. W. and Carey, S. (2008). How counting represents number: What children must learn and when they learn it. *Cognition*, 108(3):662–674. <https://doi.org/10.1016/j.cognition.2008.05.007>.
- [262] Saxena, A., Lo, C. K., Hew, K. F., and Wong, G. K. W. (2019). Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood Education. *The Asia-Pacific Education Researcher*, 29(1):55–66. <https://doi.org/10.1007/s40299-019-00478-w>.
- [263] Scaife, M., Rogers, Y., Aldrich, F., and Davies, M. (1997). Designing for or designing with? informant design for interactive learning environments. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 343–350. <https://dl.acm.org/doi/pdf/10.1145/258549.258789>.
- [264] Scherer, R., Siddiq, F., and Viveros, B. S. (2019). The cognitive benefits of learning computer programming: a meta-analysis of transfer effects. *Journal of Educational Psychology*, 111(5):764–792. <https://doi.org/10.1037/edu0000314>.
- [265] Schuler, D. and Namioka, A. (1993). *Participatory design: Principles and practices*. CRC Press.
- [266] Schwarz, C. V., Passmore, C., and Reiser, B. J. (2017). *Helping students make sense of the world using next generation science and engineering practices*. NSTA Press.

- [267] Scriven, M. (1967). The methodology of evaluation. In Tyler, R., Gagné, R., and Scriven, M., editors, *Perspectives of Curriculum Evaluation, AERA Monograph Series on Curriculum Evaluation*, pages 39–83. Rand McNally.
- [268] Seber, G. A. F. (1984). *Multivariate Observations*. Wiley. <http://doi.org/10.1002/9780470316641>.
- [269] Sedgwick, P. (2014). Multiple hypothesis testing and bonferroni’s correction. *BMJ*, 349(oct20 3):g6284–g6284. <http://doi.org/10.1136/bmj.g6284>.
- [270] Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O’Grady-Cunniff, D., Owens, B. B., Stephenson, C., and Verno, A. (2011). *CSTA K–12 Computer Science Standards: Revised 2011*. Association for Computing Machinery. <https://dl.acm.org/doi/pdf/10.1145/2593249>.
- [271] Selby, C. C. (2014). *How can the teaching of programming be used to enhance computational thinking skills?* PhD thesis, University of Southampton. <https://eprints.soton.ac.uk/366256/>.
- [272] Selby, C. C. and Woollard, J. (2013). Computational thinking: the developing definition. Technical report, University of Southampton. <https://eprints.soton.ac.uk/356481/>.
- [273] Sevin, R. and Decamp, W. (2016). From playing to programming: The effect of video game play on confidence with computers and an interest in computer science. *Sociological Research Online*, 21(3):14–23. <https://doi.org/10.5153/sro.4082>.
- [274] Shahnawazuddin, S., Adiga, N., and Kathania, H. K. (2017). Effect of prosody modification on children’s asr. *IEEE Signal Processing Letters*, 24(11):1749–1753. <https://doi.org/10.1109/LSP.2017.2756347>.
- [275] Shute, V. J., Sun, C., and Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22:142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>.
- [276] Simmering, V. R., Ou, L., and Bolsinova, M. (2019). What technology can and cannot do to support assessment of non-cognitive skills. *Frontiers in Psychology*, 10. <https://doi.org/10.3389/fpsyg.2019.02168>.
- [277] SNSF (2021). Open Science. <https://www.snf.ch/en/dah3uC2QX95tfPNd/topic/open-science>.

- [278] Soofi, A. A. and Uddin, M. (2019). A systematic review of domains, techniques, delivery modes and validation methods for intelligent tutoring systems. *International Journal of Advanced Computer Science and Applications*, 10(3). <http://doi.org/10.14569/ijacsa.2019.0100312>.
- [279] Stanja, J., Gritz, W., Krugel, J., Hoppe, A., and Dannemann, S. (2022). Formative assessment strategies for students' conceptions—the potential of learning analytics. *British Journal of Educational Technology*, 54(1):58–75. <https://doi.org/10.1111/bjet.13288>.
- [280] Stone, M. and Brooks, R. J. (1990). Continuum regression: Cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 52(2):237–258. <http://doi.org/10.1111/j.2517-6161.1990.tb01786.x>.
- [281] Sun, D., Ouyang, F., Li, Y., and Zhu, C. (2021). Comparing learners' knowledge, behaviors, and attitudes between two instructional modes of computer programming in secondary education. *International Journal of STEM Education*, 8(1). <https://doi.org/10.1186/s40594-021-00311-1>.
- [282] Sun, L., Hu, L., and Zhou, D. (2022). Programming attitudes predict computational thinking: Analysis of differences in gender and programming experience. *Computers & Education*, 181:104457. <https://doi.org/10.1016/j.compedu.2022.104457>.
- [283] Surendran, P. (2012). Technology Acceptance Model: A Survey of Literature. *International Journal of Business and Social Research*, 2(4):175–178. <https://ideas.repec.org/a/mir/mirbus/v2y2012i4p175-178.html>.
- [284] Sweeney, T., West, D., Groessler, A., Haynie, A., Higgs, B. M., Macaulay, J., Mercer-Mapstone, L., and Yeo, M. (2017). Where's the transformation? unlocking the potential of technology-enhanced assessment. *Teaching & Learning Inquiry*, 5(1):1–13. <https://doi.org/10.20343/5.1.5>.
- [285] Swider-Cios, E., Vermeij, A., and Sitskoorn, M. M. (2023). Young children and screen-based media: The impact on cognitive and socioemotional development and the importance of parental mediation. *Cognitive Development*, 66:101319. <https://doi.org/10.1016/j.cogdev.2023.101319>.

- [286] Swiss Conference of Cantonal Ministers of Education (2007). Intercantonal agreement on harmonisation of compulsory education (harmos agreement). (Accessed on 29 November 2023) <http://edudoc.ch/record/24711>.
- [287] Sysło, M. M. (2015). From algorithmic to computational thinking: On the way for computing for all students. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ITICSE '15. ACM. <https://doi.org/10.1145/2729094.2742582>.
- [288] Sysło, M. M. and Kwiatkowska, A. B. (2015). *Introducing a New Computer Science Curriculum for All School Levels in Poland*, page 141–154. Springer International Publishing. https://doi.org/10.1007/978-3-319-25396-1_13.
- [289] Tai, R. H., Ryoo, J. H., Skeeles-Worley, A., Dabney, K. P., Almarode, J. T., and Maltese, A. V. (2022). (re-)designing a measure of student's attitudes toward science: a longitudinal psychometric approach. *International Journal of STEM Education*, 9(1). <https://doi.org/10.1186/s40594-022-00332-4>.
- [290] Tall, D. (2006). A Theory of Mathematical Growth Through Embodiment, Symbolism and Proof. *ANNALES de DIDACTIQUE et de SCIENCES COGNITIVES, IREM de STRASBOURG*, 11:195 – 215. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8b6678300961be114ba653c5a5da3ab595eae0>.
- [291] Tall, D. (2013). *How Humans Learn to Think Mathematically: Exploring the Three Worlds of Mathematics*. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CB09781139565202>.
- [292] Tall, D. (2020). Making Sense of Mathematical Thinking over the Long Term: The Framework of Three Worlds of Mathematics and New Developments. *MINTUS: Beiträge zur mathematischen, naturwissenschaftlichen und technischen Bildung*. Wiesbaden: Springer. <https://homepages.warwick.ac.uk/staff/David.Tall/pdfs/dot2020a-3worlds-extension.pdf>.
- [293] Tang, X., Yin, Y., Lin, Q., Hadad, R., and Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148:103798. <https://doi.org/10.1016/j.compedu.2019.103798>.
- [294] Tedre, M. and Denning, P. J. (2016). The long quest for computational thinking. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, Koli Calling 2016. ACM. <https://doi.org/10.1145/2999541.2999542>.

- [295] Thalheim, B. (2000). The database design process. In *Entity-Relationship Modeling*, pages 13–28. Springer, Berlin, Heidelberg.
- [296] The Japan News (2016). Japan plan to make programming mandatory at schools a step to foster creativity. <https://computinged.wordpress.com/2016/07/08/>.
- [297] The Scottish Government (2016). Enhancing learning and teaching through the use of digital technology. <https://www.gov.scot/publications/enhancing-learning-teaching-through-use-digital-technology/>.
- [298] Tidwell, J. (2010). *Designing interfaces: Patterns for effective interaction design*. O'Reilly Media. <https://books.google.ch/books?id=5gv0U9X0fu0C>.
- [299] Tikva, C. and Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers & Education*, 162:104083. <https://doi.org/10.1016/j.compedu.2020.104083>.
- [300] Tsai, M. J., Liang, J. C., and Hsu, C. Y. (2020). The computational thinking scale for computer literacy education. *Journal of Educational Computing Research*, 59(4):579–602. <https://doi.org/10.1177/0735633120972356>.
- [301] Tsarava, K., Moeller, K., and Ninaus, M. (2018). Training Computational Thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, 5(2):25–44. <https://doi.org/10.17083/ijsg.v5i2.248>.
- [302] Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., and Ninaus, M. (2017). Training Computational Thinking: Game-Based Unplugged and Plugged-in Activities in Primary School. In *European conference on games based learning*, pages 687–695. Academic Conferences International Limited. https://www.researchgate.net/publication/320491120_Training_Computational_Thinking_Game-Based_Unplugged_and_Plugged-in_Activities_in_Primary_School.
- [303] Tukey, J. W. (1949). Comparing individual means in the analysis of variance. *Biometrics*, 5(2):99. <http://doi.org/10.2307/3001913>.
- [304] Tönnsen, K.-C. (2021). The relevance of trial-and-error: Can trial-and-error be a sufficient learning method in technical problem-solving-contexts? *Techne serien - Forskning i slöjdpedagogik och slöjdvetenskap*, 28(2):303–312. <https://journals.oslomet.no/index.php/techneA/article/view/4391>.

- [305] UK, G. (2013). National curriculum in england: computing programmes of study. Department for Education. [<https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>].
- [306] Ukkonen, A., Pajchel, K., and Mifsud, L. (2024). Teachers' understanding of assessing computational thinking. *Computer Science Education*, page 1–26. <https://doi.org/10.1080/08993408.2024.2365566>.
- [307] UNESCO (2024). Global education monitoring report: Slovakia | technology. <https://education-profiles.org/europe-and-northern-america/slovakia/~technology>.
- [308] UNESCO Institute for Statistics (2012). International standard classification of education: Isced 2011. *Comparative Social Research*, 30. <https://doi.org/10.15220/978-92-9189-123-8-en>.
- [309] Unnikrishnan, R., Amrita, N., Muir, A., and Rao, B. (2016). Of Elephants and Nested Loops: How to Introduce Computing to Youth in Rural India. In *Proceedings of the The 15th International Conference on Interaction Design and Children*, pages 137–146. ACM. <https://doi.org/10.1145/2930674.2930678>.
- [310] Valguarnera, S. and Landoni, M. (2023). Design with and for children: The challenge of inclusivity. In *Lecture Notes in Computer Science*, pages 171–184. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-35681-0_11.
- [311] Valguarnera, S., Sylla, C. M., and Landoni, M. (2023). The IDC research and design challenge throughout the years: achievements, reflections and next steps. In *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*. ACM. <https://doi.org/10.1145/3585088.3589382>.
- [312] Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.
- [313] Vlachogianni, P. and Tselios, N. (2021). Perceived usability evaluation of educational technology using the system usability scale (sus): A systematic review. *Journal of Research on Technology in Education*, 54(3):392–409. <https://doi.org/10.1080/15391523.2020.1867938>.
- [314] Vomlel, J. (2004). Building adaptive tests using bayesian networks. *Kybernetika*, 40(3):[333]–348. <http://eudml.org/doc/33704>.

- [315] Voogt, J., Fisser, P., Good, J., Mishra, P., and Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and information technologies*, 20:715–728. <https://doi.org/10.1007/s10639-015-9412-6>.
- [316] Voogt, J. and Roblin, N. P. (2012). A comparative analysis of international frameworks for 21st century competences: Implications for national curriculum policies. *Journal of Curriculum Studies*, 44(3):299–321. <https://doi.org/10.1080/00220272.2012.668938>.
- [317] Voronina, L. V., Sergeeva, N. N., and Utyumova, E. A. (2016). Development of algorithm skills in preschool children. *Procedia - Social and Behavioral Sciences*, 233:155–159. <https://doi.org/10.1016/j.sbspro.2016.10.176>.
- [318] Vujičić, L., Jančec, L., and Mezak, J. (2021). Development of algorithmic thinking skills in early and preschool education. In *EDULEARN21 Proceedings*, 13th International Conference on Education and New Learning Technologies, pages 8152–8161. IATED. <http://doi.org/10.21125/edulearn.2021.1650>.
- [319] Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes.
- [320] Wahyuningsih, S., Nurjanah, N., Rasmani, U., Hafidah, R., Pudyaningtyas, A., and Syamsuddin, M. (2020). Steam learning in early childhood education: A literature review. *International Journal of Pedagogy and Teacher Education*, 4(1):33–44. <http://doi.org/10.20961/ijpte.v4i1.39855>.
- [321] Wallet, P. et al. (2015). *Information and Communication Technology (ICT) in Education in sub-Saharan Africa: A comparative analysis of basic e-readiness in schools*. Information Paper No. 25. UNESCO Institute for Statistics. <https://doi.org/10.15220/978-92-9189-178-8-en>.
- [322] Wang, J. and Hejazi Moghadam, S. (2017). Diversity barriers in k-12 computer science education: Structural and social. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17. ACM. <https://doi.org/10.1145/3017680.3017734>.
- [323] Wang, M.-T., Guo, J., and Degol, J. L. (2019). The role of sociocultural factors in student achievement motivation: A cross-cultural review. *Adolescent Research Review*, 5(4):435–450. <https://doi.org/10.1007/s40894-019-00124-y>.

- [324] Wang, X., Dai, M., and Mathis, R. (2022b). The influences of student- and school-level factors on engineering undergraduate student success outcomes: A multi-level multi-school study. *International Journal of STEM Education*, 9(1). <http://doi.org/10.1186/s40594-022-00338-y>.
- [325] Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., and Sysło, M. M. (2017). Computer science in k-12 school curricula of the 21st century: Why, what and when? *Education and Information Technologies*, 22(2):445–468. <https://doi.org/10.1007/s10639-016-9493-x>.
- [326] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., and Wilensky, U. (2015). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1):127–147. <https://doi.org/10.1007/s10956-015-9581-5>.
- [327] Weintrop, D., Rutstein, D. W., Bienkowski, M., and McGee, S. (2021). Assessing computational thinking: an overview of the field. *Computer Science Education*, 31(2):113–116. <https://doi.org/10.1080/08993408.2021.1918380>.
- [328] Werner, L., Campe, S., and Denner, J. (2012). Children learning computer science concepts via alice game-programming. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 427–432.
- [329] Wickey da Silva Garcia, F., Bezerra Oliveira, S. R., and da Costa Carvalho, E. (2022). Application of a teaching plan for algorithm subjects using active methodologies: An experimental report. *International Journal of Emerging Technologies in Learning (Ijet)*, 17(07):175–207. <https://doi.org/10.3991/ijet.v17i07.28733>.
- [330] Wilkerson, M. H. and Fenwick, M. (2017). Using mathematics and computational thinking. *Helping students make sense of the world using next generation science and engineering practices*, pages 181–204.
- [331] Williges, R. (1984). Evaluating human-computer software interfaces. In *Proceedings of International Conference on Occupational Ergonomics*, pages 81–87.
- [332] Wilpon, J. and Jacobsen, C. (1996). A study of speech recognition for children and the elderly. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 349–352. <https://doi.org/10.1109/ICASSP.1996.541104>.

- [333] Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212. <http://doi.org/10.1080/01621459.1927.10502953>.
- [334] Wing, J. (2017). Computational thinking’s influence on research and education for all. *Italian Journal of Educational Technology*, 1(1). <https://doi.org/10.17471/2499-4324/922>.
- [335] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35. <https://doi.org/10.1145/1118178.1118215>.
- [336] Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3717–3725. <https://doi.org/10.1098/rsta.2008.0118>.
- [337] Wing, J. M. (2011). Research notebook: Computational thinking—what and why. *The link magazine*, 6:20–23. <https://people.cs.vt.edu/~kafura/CS6604/Papers/CT-What-And-Why.pdf>.
- [338] Wing, J. M. (2014). Computational thinking benefits society. *40th anniversary blog of social issues in computing*, 2014:26. <http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>.
- [339] Wohl, B., Porter, B., and Clinch, S. (2015). Teaching Computer Science to 5-7 year-olds: An initial study with Scratch, Cubelets and unplugged computing. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, pages 55–60. ACM. <https://doi.org/10.1145/2818314.2818340>.
- [340] Wu, L. (2019). Student model construction of intelligent teaching system based on bayesian network. *Personal and Ubiquitous Computing*, 24(3):419–428. <https://doi.org/10.1007/s00779-019-01311-3>.
- [341] Xing, W., Li, C., Chen, G., Huang, X., Chao, J., Massicotte, J., and Xie, C. (2020). Automatic assessment of students’ engineering design performance using a bayesian network model. *Journal of Educational Computing Research*, 59(2):230–256. <http://doi.org/10.1177/0735633120960422>.
- [342] Yadav, A., Mayfield, C., Zhou, N., Hambruch, S. E., and Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *Acm Transactions on Computing Education*, 14(1):1–16. <https://doi.org/10.1145/2576872>.

- [343] Yarosh, S., Radu, I., Hunter, S., and Rosenbaum, E. (2011). Examining values: an analysis of nine years of idc research. In *Proceedings of the 10th International Conference on Interaction Design and Children*. ACM. <https://doi.org/10.1145/1999030.1999046>.
- [344] Yates, F. (1934). Contingency tables involving small numbers and the χ^2 test. *Supplement to the Journal of the Royal Statistical Society*, 1(2):217. <http://doi.org/10.2307/2983604>.
- [345] Yeung, G. and Alwan, A. (2018). On the difficulties of automatic speech recognition for kindergarten-aged children. In *Interspeech 2018*. ISCA. <https://doi.org/10.21437/interspeech.2018-2297>.
- [346] Yildiz, C. D. (2020). Ideal classroom setting for english language teaching through the views of english language teachers (a sample from turkey). *English Language Teaching*, 13(3):31. <https://doi.org/10.5539/elt.v13n3p31>.
- [347] Zapata-Cáceres, M. and Fanchamps, N. (2021). Using the Beginners Computational Thinking Test to Measure Development on Computational Concepts Among Preschoolers. In *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Conference 2021*, pages 32–37, Taiwan, Province of China. Asia-Pacific Society for Computers in Education. <https://cte-stem2021.nie.edu.sg/>, https://cte-stem2021.nie.edu.sg/assets/docs/CTE-STEM_Compiled-Proceedings.pdf.
- [348] Zapata-Cáceres, M., Martín-Barroso, E., and Román-González, M. (2020). Computational Thinking Test for Beginners: Design and Content Validation. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 1905–1914. <https://doi.org/10.1109/EDUCON45650.2020.9125368>.
- [349] Zdaniuk, B. (2014). *Ordinary Least-Squares (OLS) Model*, page 4515–4517. Springer Netherlands. http://doi.org/10.1007/978-94-007-0753-5_2008.

Research contributions

Scientific publications

Peer-reviewed

- [A1.1] Adorni, G., Artico, I., Piatti, A., Lutz, E., Gambardella, L. M., Negrini, L., Mondada, F., and Assaf, D. (2024a). Development of algorithmic thinking skills in K-12 education: A comparative study of unplugged and digital assessment instruments. *Computers in Human Behavior Reports*, 15:100466. <https://doi.org/10.1016/j.chbr.2024.100466>.
- [A1.2] Adorni, G., Mangili, F., Piatti, A., Bonesana, C., and Antonucci, A. (2023). Rubric-based Learner Modelling via Noisy Gates Bayesian Networks for Computational Thinking Skills Assessment. *Journal of Communications Software and Systems*, 19(1):52–64. <http://doi.org/10.24138/jcomss-2022-0169>.
- [A1.3] Adorni, G., Piatti, A., Bumbacher, E., Negrini, L., Mondada, F., Assaf, D., Mangili, F., and Gambardella, L. (2025). FADE-CTP: A Framework for the Analysis and Design of Educational Computational Thinking Problems. *Technology, Knowledge and Learning*. <https://doi.org/10.1007/s10758-025-09833-x>.
- [A1.4] Adorni, G., Piatti, S., and Karpenko, V. (2024b). Virtual CAT: A multi-interface educational platform for algorithmic thinking assessment. *SoftwareX*, 27:101737. <https://doi.org/10.1016/j.softx.2024.101737>.
- [A1.5] Antonucci, A., Mangili, F., Bonesana, C., and Adorni, G. (2021). A New Score for Adaptive Tests in Bayesian and Credal Networks. In *Vejnarová, J., Wilson, N. (eds) Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 12897, pages 399–412. Springer, Cham. ECSQARU 2021. Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-030-86772-0_29.

- [A1.6] Antonucci, A., Mangili, F., Bonesana, C., and Adorni, G. (2022). Intelligent Tutoring Systems by Bayesian Nets with Noisy Gates. *The International FLAIRS Conference Proceedings*, 35. <https://doi.org/10.32473/flairs.v35i.130692>.
- [A1.7] Corecco, S., Adorni, G., and Gambardella, L. M. (2023). Proximal Policy Optimization-Based Reinforcement Learning and Hybrid Approaches to Explore the Cross Array Task Optimal Solution. *Machine Learning and Knowledge Extraction*, 5(4):1660–1679. <http://doi.org/10.3390/make5040082>.
- [A1.8] Mangili, F., Adorni, G., Piatti, A., Bonesana, C., and Antonucci, A. (2022). Modelling Assessment Rubrics through Bayesian Networks: a Pragmatic Approach. In *2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE. <https://doi.org/10.23919/softcom55329.2022.9911432>.
- [A1.9] Piatti, A., Adorni, G., El-Hamamsy, L., Negrini, L., Assaf, D., Gambardella, L., and Mondada, F. (2022). The CT-cube: A framework for the design and the assessment of computational thinking activities. *Computers in Human Behavior Reports*, 5:100166. <https://doi.org/10.1016/j.chbr.2021.100166>.

Preprints (under review)

- [A2.1] Adorni, G. and Piatti, A. (2024). Designing the virtual CAT: A digital tool for algorithmic thinking assessment in compulsory education. Under review at the International Journal of Child-Computer Interaction. <https://doi.org/10.48550/arXiv.2408.01263>.
- [A2.2] Assaf, D., Adorni, G., Lutz, E., Negrini, L., Piatti, A., Mondada, F., Mangili, F., and Gambardella, L. M. (2024). The CTSkills App – Measuring Problem Decomposition Skills of Students in Computational Thinking. <https://doi.org/10.48550/arXiv.2411.14945>.

Datasets

- [B.1] Adorni, G. (2023). Dataset for algorithmic thinking skills assessment: Results from the virtual CAT pilot study in Swiss compulsory education (1.0.0). Zenodo Dataset. <https://doi.org/10.5281/zenodo.10018292>.

- [B.2] Adorni, G. (2024). Dataset for algorithmic thinking skills assessment: Results from the virtual CAT large-scale study in Swiss compulsory education (1.0.0). Zenodo Dataset. <https://doi.org/10.5281/zenodo.10912339>.

Other research outputs (software, protocols, and materials)

- [C.1] Adorni, G. (2024). Virtual CAT Algorithmic Thinking Assessment: Data Analysis Procedures. Zenodo Software. <https://doi.org/10.5281/zenodo.12805318>.
- [C.2] Adorni, G. and Bonesana, C. (2023). BN-based Learning Networks with Noisy Gates (1.0.0). Zenodo Software. <https://doi.org/10.5281/zenodo.10958613>.
- [C.3] Adorni, G. and Karpenko, V. (2023a). virtual CAT data infrastructure (1.0.0). Zenodo Software. <https://doi.org/10.5281/zenodo.10015011>.
- [C.4] Adorni, G. and Karpenko, V. (2023b). virtual CAT programming language interpreter (1.0.0). Zenodo Software. <https://doi.org/10.5281/zenodo.10016535>.
- [C.5] Adorni, G., Piatti, S., and Karpenko, V. (2023). virtual CAT: An app for algorithmic thinking assessment within Swiss compulsory education (1.0.1). Zenodo Software. <https://doi.org/10.5281/zenodo.10027851>.
- [C.6] Corecco, S. and Adorni, G. (2023). CAT Optimal Hybrid Solver (1.0.0). Zenodo Software. <https://doi.org/10.5281/zenodo.8375222>.
- [C.7] Piatti, A. and Adorni, G. (2024). Unplugged Cross Array Task (CAT) Assessment: Supplementary Documentation and Experimental Protocol. Zenodo. <https://doi.org/10.5281/zenodo.12806226>.