Therapists as designers: an initial investigation of end-user programming of a tangible tool for therapeutic interventions

Margherita Andrao [1,2], Barbara Treccani [1] and Massimo Zancanaro [1,2]

[1] *Department of Psychology and Cognitive Science, University of Trento, Rovereto (Trento), Italy*
[2] *Fondazione Bruno Kessler – FBK, Trento, Italy*

Abstract

This paper presents a pilot study on end-user programming by therapists of a tangible tool for children on the autism spectrum. The core design ideas were to use detailed natural language descriptions of states and events, and an incremental process to facilitate the programming task. Our study provides initial evidence of the feasibility of this approach.

Keywords

End-User Programming, End-User Development, Mental Models

1. Introduction

This study aimed to shed new light on end-user programming by naïve users and -in particular- on how users develop and assume appropriate mental models for programming tasks. Specifically, our work was meant to investigate whether a proper representation of the task and the use of a concrete representation of events and states may facilitate learning of an EUD tool. To this aim, we examined a group of therapists while they were learning to program a tangible table to support individual exercises for children on the autism spectrum. We used a *thinking-aloud* procedure [2] to investigate therapists' mental representations of how to program this tool.

End-User Development (EUD) is the possibility for naïve users to create and modify computer applications [11]. EUD is a valuable approach to accommodate users' idiosyncratic needs and allow both the expression of users' creativity and the attribution of personal values and meanings to artifacts [12]. When it comes to social therapy for children on the autism spectrum, these two aspects are tightly interconnected. Indeed, it is well known that children on the autism spectrum have idiosyncratic and strong dispositions that need to be taken into consideration by therapists for the success of the intervention [8]. Various approaches have been used to support therapists for personalizing ICT-based interventions while the use of EUD has been proposed [6] but not yet thoughtfully explored.

In this study, we built upon the notion of Trigger-Action rules for controlling IoT devices [1,14] and leveraged on the idea of communicating a clear distinction between events and states [see 3,4] to propose a language of primitives that therapists can use to personalize the exercises in a tangible device [15].

1.1. The tangible tool and the events/state primitives

In order to present a realistic example that could be also used in further longitudinal studies, in this pilot study we referred to the tabletop tangible tool described in Wierbiłowicz and colleagues [15]. This tool is an IoT device used to present interactive, multimedia, and tangible classification exercises. It has

EMPATHY: Empowering People in Dealing with Internet of Things Ecosystems. Workshop co-located with INTERACT 2021, August 30, 2021, Bari, Italy
EMAIL: margherita.andrao@unitn.it (M. Andrao); barbara.treccani@unitn.it (B. Treccani); massimo.zancanaro@unitn.it (M. Zancanaro)
ORCID: 0000-0003-2245-9835 (M. Andrao); 0000-0001-8028-0708 (B. Treccani); 0000-0002-1554-5703 (M. Zancanaro)

a circular surface (approximately 50 cm) carved with guides for the placement of the external pieces: objects and tiles representing different values of the objects' properties (see Figure 1). In the study by Wierbiłowicz and collaborators [15], this tool was used for classification exercises. This type of exercise starts when the therapist places an object in the center of the table. The child has to select the tiles that correspond to the specific values (e.g., “red”, “large”, “circular”) of the properties (e.g., color, size, shape) of this object. Once the tiles are placed in the guides, the table recognizes the objects and tiles through NFC tags and displays proper visual and acoustical feedback in case of either success or error. As discussed by the authors, the combination of digital and materiality has some interesting effects on sustained attention and empowerment of both the therapists and the children. In assessing their experience with the table, the therapists in Wierbiłowicz et al.’s study expressed the need to personalize the type and form of the feedback to adapt the exercises to any specific child.

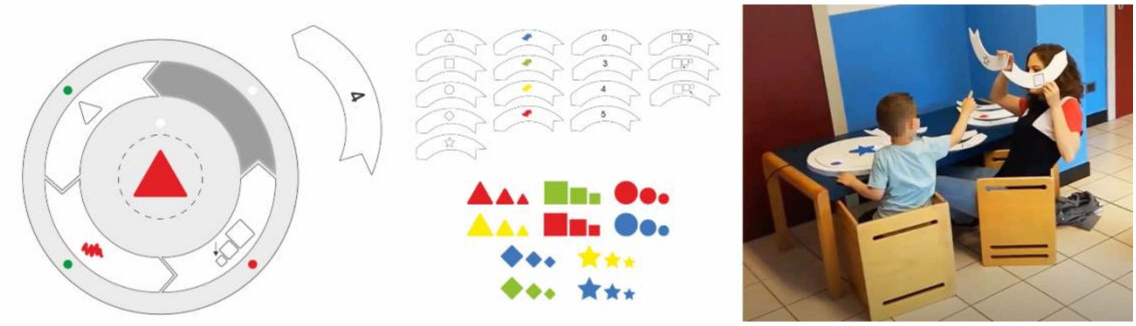

Figure 1: The tabletop tangible tool (left) with tiles and objects (middle), and a photo of a therapist using the table with a child (left) from Wierbiłowicz and colleagues [15].

1.1.1. Primitive for states and events

In order to ground the EUD tasks, primitives for states and events were developed to reflect actual configurations on the table and actions that users can perform on the table, respectively. The action parts of the trigger-action rules are described as commands that can be given on the table to control lights and sounds. To keep the task simple for non-programmers, we opted to avoid the use of variables [13], while relying on lengthy and redundant descriptions of events and state (this should alleviate the selection and the information barriers described by Ko and colleagues [9]). Natural language descriptions have been demonstrated to be useful for programming task comprehension [5], although they might be confusing when the procedures need to be generated [7]. In an attempt to overcome this problem, we employed a constrained interface that can guide the users in composing each rule by progressively selecting the events and state for each rule (see Figure 2; the actual interface has been inspired by the one proposed by Desolda and colleagues [3]). An example of a state description is “[WHILE] at least 2 tiles in the lateral areas conform to the [properties of] the central object”. Similarly, an example of an event is “[WHEN] an object is inserted in the central place”. An example of an action with a contextual dependency (to avoid the use of a variable) is “switch off the light of the last tile added to the table”. In total, 90 state descriptions, 684 event descriptions, and 368 action specifications have been defined. The graphical user interface is designed to support the user in progressively build the statements.

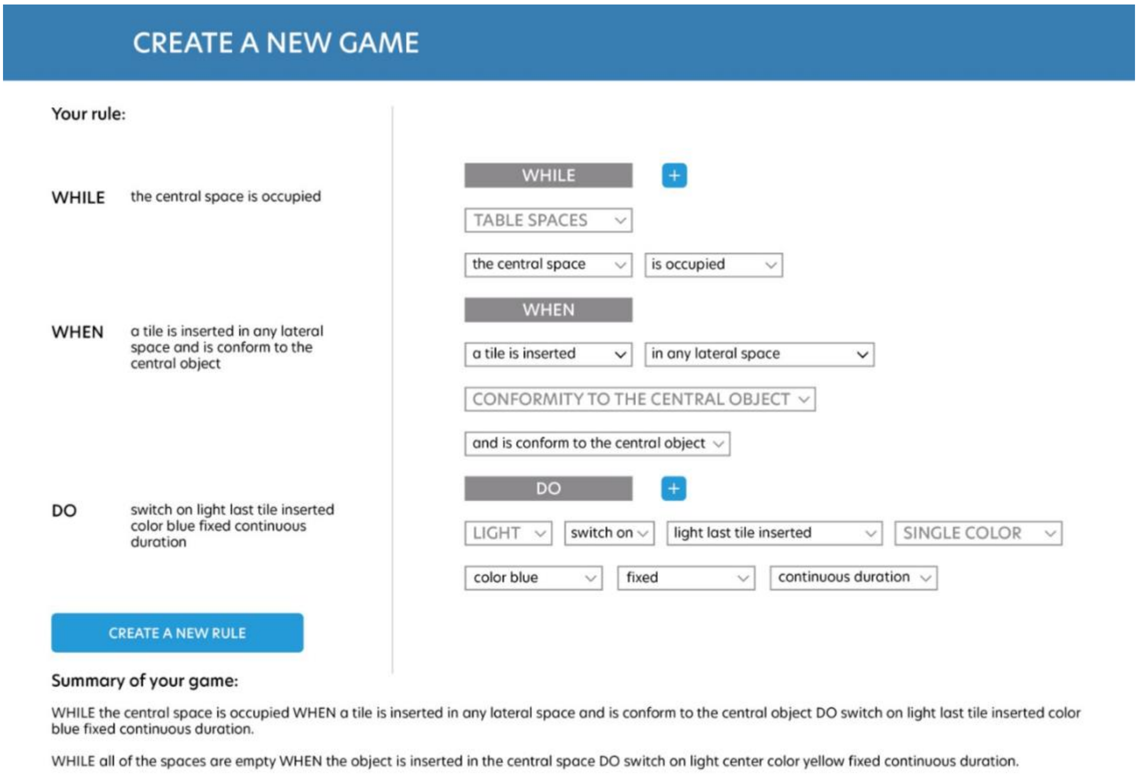

Figure 2: A screenshot of the interface used for the study. The central area contains the form to guide the construction of the specifications of the Trigger-Action Rules. The lengthy verbose descriptions of events and states make the rules look similar to natural language statements.

2. The study

Four (4) volunteers (1 male, 3 females) were recruited for this pilot study. The inclusion criteria were work experience as a psychologist, therapist or educator in the context of cognitive, behavioral and social rehabilitation with children on the autism spectrum as well as no previous experience with programming. The study has been conducted remotely as semi-structured individual interviews. Each session was about 40 minutes and started with a brief demonstration of both the tangible tool and the EUD graphical interface. Each participant then was asked to perform three simple 1-rule tasks to familiarize with the system and with the think-aloud protocol. Finally, s/he had to create a new type of exercise for a classification game. Participants were instructed on the mechanics of the game while they have to create the rules using the available primitives. At the end of the study, participants were debriefed and asked about their comments, impressions, and possibly advices to improve the system.

2.1. Preliminary results

A managed to correctly define most of the rules. Participants B and C were able to correctly define almost all the rules. Participants A and B described each single configuration rather than using the more abstract state descriptions. Participants C was able to use oftentimes the more general (abstract) statements. Participant D was not able to correctly frame the problem with respect to the rules. Participant A always confused “while” and “when” conditions, but eventually he was able to correct himself most of the times by reading the rules’ summary. Indeed, the incremental rule creation through sequences of drop lists seemed for him to be more confusing than helpful while the final summary of the rules (which he read as a natural language statement) was much useful to recognize the mistakes.

Instead, Participants B and C took advantage of the incremental construction of the interface. They almost "reasoned through the interface" rather than reasoning on the task before and then entering the rules into the interface. They too sometimes confused while with when but they were able to use these two conditions correctly while building the rule in the interface. Participant D used the strategy of thinking on the task before looking into the interface, similarly to participant A, and she too usually confused events with states, for example she complained when she could not find an event like "when all the tiles are inserted …". Differently than participant A, she was not able to take advantage of the natural language descriptions.

3. Conclusion

Although very preliminary, our results are consistent with the insights from previous literature [5,7] on the effectiveness of natural language descriptions on program's comprehension. Furthermore, our study suggests that an incremental interface may also help the users in the generation phase, but only if they use the interface while reasoning on the task. Finally, it is worth noting that this approach may effectively facilitate reasoning on concrete configurations (as done by participants B and C) but it does not seem to help abstract reasoning. This might be a major weakness of the approach. Further studies are needed to confirm the value of this approach and thoroughly investigate its pros and cons.

4. Acknowledgements

This work has been supported by the Italian Ministry of Education, University and Research (MIUR) under grant PRIN 2017 "EMPATHY: EMpowering People in deAling with internet of THings ecosYstems" (Progetti di Rilevante Interesse Nazionale – Bando 2017, Grant 2017MX9T7H)

5. References

[1] C. Ardito et al., User-defined semantics for the design of IoT systems enabling smart interactive experiences, Pers. Ubiquitous Comput. 24 (2020) 781–796, doi:10.1007/s00779-020-01457-5.

[2] T. Boren and J. Ramey, Thinking aloud: reconciling theory and practice, IEEE Trans. Prof. Commun 43 (2000) 261–278, doi: 10.1109/47.867942.

[3] G. Desolda, F. Greco, F. Guarnieri, N. Mariz, and M. Zancanaro, SENSATION: An Authoring Tool to Support Event–State Paradigm in End-User Development, in: C. Ardito, R. Lanzilotti, A. Malizia, H. Petrie, A. Piccinno, G. Desolda, and K. Inkpen (Eds.), Human-Computer Interaction – INTERACT 2021, vol. 12933, Cham: Springer International Publishing, 2021, pp. 373–382, doi: 10.1007/978-3-030-85616-8_22.

[4] G. Gallitto, B. Treccani, and M. Zancanaro, If when is better than if (and while might help): on the importance of influencing mental models in EUD (a pilot study), in: Proceedings of the 1st International Workshop on Empowering People in Dealing with Internet of Things Ecosystems - co-located with International Conference on Advanced Visual Interfaces (AVI 2020), Island of Ischia, Italy, Sep. 2020, pp. 7-11.

[5] K. M. Galotti and W. F. Ganong, What non-programmers know about programming: Natural language procedure specification, Int. J. Man-Mach. Stud. 22 (1985) 1–10, doi: 10.1016/S0020-7373(85)80073-0.

[6] F. Garzotto and R. Gonella, An open-ended tangible environment for disabled children's learning, in: Proceedings of the 10th International Conference on Interaction Design and Children, New York, NY, USA, Jun. 2011, pp. 52–61, doi: 10.1145/1999030.1999037.

[7] J. Good and K. Howland, Programming language, natural language? Supporting the diverse computational activities of novice programmers, J. Vis. Lang. Comput. 39 (2017) 78–92, doi: 10.1016/j.jvlc.2016.10.008.

[8] C. Harrop, J. Amsbary, S. Towner-Wright, B. Reichow, and B. A. Boyd, That's what I like: The use of circumscribed interests within interventions for individuals with autism spectrum disorder. A systematic review, Res. Autism Spectr. Disord. 57 (2019) 63–86, doi: 10.1016/j.rasd.2018.09.008.
[9] A. J. Ko, B. A. Myers, and H. H. Aung, Six Learning Barriers in End-User Programming Systems, in: Proceedings of 2004 IEEE Symposium on Visual Languages - Human Centric Computing, Rome, 2004, pp. 199–206. doi: 10.1109/VLHCC.2004.47.
[10] J. Korte et al., Pushing the Boundaries of Participatory Design, in: Proceedings of Human-Computer Interaction – INTERACT 2019, Paphos, Cyprus, Sep. 2019, pp. 747–753. doi: 10.1007/978-3-030-29390-1_74.
[11] H. Lieberman, F. Paternò, M. Klann, and V. Wulf, End-User Development: An Emerging Paradigm, in: H. Lieberman, F. Paternò, and V. Wulf (Eds.), End User Development, vol. 9, Dordrecht: Springer Netherlands, 2006, pp. 1–8. doi: 10.1007/1-4020-5386-X_1
[12] P. Markopoulos, J. Nichols, F. Paternò, and V. Pipek, Editorial: End-User Development for the Internet of Things, ACM Trans. Comput.-Hum. Interact. 24, (2017), 1–3, doi: 10.1145/3054765
[13] J. F. Pane, C. A. Ratanamahatana, and B. A. Myers, Studying the language and structure in non-programmers' solutions to programming problems, Int. J. Hum.-Comput. Stud. 54 (2001) 237–264, doi: 10.1006/ijhc.2000.0410.
[14] F. Paternò and C. Santoro, End-user development for personalizing applications, things, and robots, Int. J. Hum.-Comput. Stud. 131 (2019) 120–130, doi: 10.1016/j.ijhcs.2019.06.002.
[15] J. Wierbiłowicz et al., Look at me and grab this! Materiality and the practices around negotiation of social attention with children on the autistic spectrum, in: Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society, New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–5, doi: https://doi.org/10.1145/3419249.3420176