End-User Programming for Smart Educational Devices: Exploring the Role of Linguistic Aspects, Mental Models, and Reasoning Strategies in Trigger-Action Rules Composition and Debugging

Margherita Andrao [1,2]

[1] University of Trento, Trento, Italy
[2] Fondazione Bruno Kessler, Trento, Italy

Abstract

Enabling individuals to personalize and control the functionality of their smart devices is essential, particularly in educational environments where customized curricula are necessary for each student. To achieve this goal, End-User Programming through trigger-action rules seems to be a promising approach to empower teachers to reach this goal. However, to handle complex scenarios, End-User Programming systems must support naive users in adopting effective reasoning strategies and mental models. My Ph.D. research intends to explore specific linguistic aspects that can guide teachers in creating and debugging trigger-action rules for programming their smart educational devices, supporting them to assume effective mental models and reasoning strategies.

Keywords

End-user programming, Linguistic Aspects, Cognitive Processes, Educational Technologies

1. Introduction

The application of smart devices in education, such as the Internet of Things (IoT), is gaining increasing attention for their well-known potential to enhance and support various aspects of the learning-teaching process [1, 2, 3]. Interactive educational devices were designed and applied to multiple school contexts, for example, music [4], math, and language (e.g., [5, 6]). IoT educational devices provide an interactive and engaging learning environment [1, 7], allowing students to build their own knowledge (see [8]) while providing teachers with constantly updated data and information on learners and their achievements [1, 2]. These aspects could promote the personalization of curricula based on individual student abilities [9], which is crucial in the learning process, particularly for those with specific learning disorders [10].

However, this research field still has several challenges to overcome. Some authors (e.g., [9, 11]) highlighted that one gap to fill is related to teachers/educators sometimes limited computer and programming skills. Since they are the primary experts in the educational field, teachers should be able to directly modify/program educational smart devices to create purpose-specific environments and integrate these devices optimally into education strategies and plans [12, 13]. Thus, one of the current challenges is enabling educators to customize their smart devices according to their educational goals, creativity, and students' needs. Therefore, it becomes essential to study and design methods and tools to allow them to reach this purpose.

This topic perfectly fits in End-User Development (EUD), an emerging field aiming to empower users who are not programming experts to create and personalize their computer applications [14, 15]. End-User Programming (EUP), in particular, intends to allow non-programmer users to define and create programs that automate behaviors of their digital artifacts [16, 17]. Among others, Trigger-Action Programming (TAP) is one of the relevant approaches in EUP (see [18]), which allows users to define rules for triggering a specific action when an event occurs [19, 20, 21, 22]. These rules are

IS-EUD 2023: 9th International Symposium on End-User Development, 6-8 June 2023, Cagliari, Italy
EMAIL: margherita.andrao@unitn.it (M. Andrao)
ORCID: 0000-0003-2245-9835 (M. Andrao)

typically presented in the form of *If* <trigger> *Then* <action>, as visible in popular automation platforms (e.g., IFTTT or Zapier [23]). The popularity and simplicity of TAP make it appropriate and suitable for individuals with no programming experience [22, 24, 25].

Yet, it has several limitations, such as its specific suitability for rules having just one event as a trigger [21, 22]. Combining multiple triggers is crucial for allowing users to create more sophisticated and expressive programs, as shown by several studies [19, 21, 22, 24, 26, 27, 28]. Indeed, TAP with multiple triggering conditions tends to approximate the Event-Condition-Action (ECA) paradigm used by expert programmers (e.g., [29, 30]). According to some studies (and in particular [21, 22, 28]), end-users, regardless of their experience in programming, managed to create programs with multiple triggers and actions in separate rules. However, several difficulties may arise when combining multiple and complex triggering conditions because of naive users' incomplete or incorrect mental models of the system, leading to ambiguities and errors in rule composition, interpretation, and debugging [20, 21].

When users interact with a particular system, they rely on their mental representation of the system's structure, operations, and functions, which is stored in long-term memory (i.e., conceptual model), to create mental models of the ongoing interaction in their working memory [31, 32, 33]. Mental models are cognitive frameworks resulting from declarative, procedural, and inferential knowledge integrated with information acquired through perceptual processes [31]. Mental models play a crucial role in human comprehension, reasoning, and cognitive processes allowing outcomes prediction of different scenarios, and faulty mental models can lead to errors in thinking (e.g., [34]).

Both research fields of psychology of programming and EUD have explored aspects that influence the creation and assumption of users' mental models. Well-designed systems should implicitly induce effective conceptual models, as Norman stated [33, 35], but explicit information may become essential as the systems increase in complexity. Indeed, some studies suggested that providing detailed information on the operations and functioning of the system helped users create accurate conceptual models, resulting in more effective interactions and precise outcome predictions (e.g., [27, 28, 36, 37]).

Particularly in the context of multiple trigger conjunctions in trigger-action rules composition, temporal features of events influence comprehension, interpretation, and the creation of effective and accurate mental models, as highlighted by Huang and Cakmak [21]. The authors identified two classes of events for trigger-action rules based on their temporal aspects: instantaneous occurrences (simply events) and protracted ones (states). However, they found that users often confused them. Indeed, the *If* <trigger> *Then* <action> form commonly used in trigger-action rules may not clearly support the distinction between these two types of occurrences. To address this issue and help naive users assume effective mental models, they suggested investing interface-level solutions, such as grouping and clearly naming states and events, using different temporal conjunctions and verbal structures to highlight events and states differences. Overall, the authors stress the importance of developing effective strategies for communicating a clear and categorical distinction between occurrences based on their temporal features.

Language use implicitly supports the distinction between temporal aspects of states and events [38, 39, 40]. Some recent studies have successfully assessed the use of specific temporal conjunctions, such as "*while*" and "*when*" to facilitate the understanding of the difference between states and events in trigger-action rules [20, 21, 41]. Another linguistic aspect to consider is the syntactic order of the elements presented, which influences individuals' mental models [40]. For example, the iconicity assumption (e.g., [42]) suggests that when the order of clauses in a sentence corresponds to the actual order of events is more easily represented by individuals. Simultaneous events (which violates iconicity assumptions) are more easily understood when the temporal conjunction indicating simultaneity is at the beginning of the sentence (e.g., "*while I am at the office, the phone ring*" vs. "*the phone ring while I am at the office"*). Similarly, Pane et al. [43] found that non-expert users naturally and consistently wrote the conjunction "*if*" or "*while*" at the beginning of the statements following the logical structure of trigger-action rules.

Therefore, it is fundamental to consider linguistic aspects when creating interfaces and choosing appropriate verbal primitives in a TAP system [44, 45]. Blackwell [46] proposed naive users' difficulties in using EUP systems emerge from the absence of direct control over the elements used in the rules and the use of abstract notational elements. Many issues and bugs in EUP may arise due to the significant gap between the mental models of users and the programming languages used [47], along with users' tendency to apply language knowledge to programming tasks [48].

In this respect, research findings on human reasoning and mental models revealed that content (in our case, verbal primitives for composing trigger-action rules) significantly influences how individuals reason and draw conclusions [49]. Some authors emphasized at least two reasoning strategies related to the creation of mental models [50, 51]: (i) concrete reasoning builds on real-world knowledge and perceptual aspects related to tangible case-specific objects for solving problems; (ii) abstract reasoning implies pattern recognition and generalization from apparently specific contexts for identifying the high-level general aspects and principles. Abstraction is fundamental in individuals' learning processes and mental models' creation [51].

Even if abstract reasoning may be more difficult for individuals than concrete reasoning, it seems to be among those cognitive tools essential in programming [52]. Indeed, it is a primary aspect of dealing with complexity [46], understanding, and problem-solving [53]. Abstract reasoning is also linked to computational thinking (CT) - a construct widely studied in education and computer science and defined as a thinking style that involves expressing and solving problems as a computer [54]. Abstraction is one of the skills of CT that supports and enables novice programmers to succeed with programming tasks [52]. As analyzed and emphasized in a recent work [55], the intersection between CT and EUD research fields could lead to new engaging directions for EUD fostering the implementation of the EUDability construct besides the usability one. Indeed, future studies and applications should deeply investigate and implement the design of EUD tools that consider and support the acquisition/development/training of skills related to CT (for example, abstraction).

In conclusion, we suggest that studying linguistics aspects, mental models, and reasoning strategies in depth could shed light on new opportunities in EUP, specifically considering systems designed for people highly motivated to personalize and control their technological devices, such as teachers.

2. The Ph.D. Project

From a more general perspective, my Ph.D. research aims to study and implement some specific linguistic aspects to support and guide naive users - specifically teachers and educators - in assuming effective mental models and reasoning strategies during interaction with event-driven systems, such as TAP environments. Several studies underlined the usefulness and effectiveness of language descriptions for the comprehension of programming tasks (e.g., [56]), the creation of effective and distinct mental models for states and events (e.g., [21]), and the assumption of reasoning strategies. A practical goal is to design and implement a textual interface based on specific linguistic aspects suitable for teachers to allow them a richer expressiveness during EUP tasks. As pointed out in the introduction, it is essential to design EUP systems to optimize the integration of smart devices in teaching and allow educators to benefit from their potential in teaching-learning processes.

Overall, my Ph.D. project can be summarized into two macro objectives. The first investigates different linguistic aspects that could influence the composition, comprehension, and interpretation of trigger-action rules and help naive users develop/assume more appropriate mental models. The second one studies the implementation of these linguistic aspects in a system designed to enable teachers to personalize their digital devices and explore mental models and reasoning strategies (e.g., abstract or concrete) during EUP tasks (specifically, composition and debugging trigger-action rules) and real interaction with their smart devices.

The achievement of these goals will contribute to the main theoretical outcome of my Ph.D. project: the development of a conceptual framework on specific linguistic aspects in EUP in relation to reasoning strategies and mental models of naive users.

2.1. Objective 1: Language in End-User Programming

The first objective aims to understand the role of some specific linguistic aspects in the distinct and more effective mental representation of occurrences with different temporal features in TAP. Specifically, we are interested in investigating if some temporal conjunctions and verbal structures are more fitted for describing states or events and if the order of clauses (containing state, event, or action) influences naive users' comprehension and mental models' assumption during EUP tasks. Specifically, we focused on three research questions:

RQ1. Do specific language features, such as conjunctions, verbal structures, and the order of elements (states, events, and actions), facilitate the creation of effective and distinct mental models for states and events in non-expert users during the composition and interpretation of trigger-action rules?

RQ2. Are the temporal conjunctions “*if*”, “*while*”, “*as long as*”, “*when*”, and “*as soon as*”, and some specific verbal structures naturally associated with the definition of states and events?

RQ3. Is there an action-state-event order naturally used by individuals that can facilitate them in composing and mentally representing trigger-action rules?

To answer these research questions, we designed three studies:

Study1A. A qualitative and explorative study on mental models and linguistic preferences among temporal conjunctions, verbal structures, and the order of clauses in trigger-action rules composition.

Study1B. An online experiment investigating linguistics preferences for temporal conjunctions and verbal structures for states and events in trigger-action rules.

Study1C. An online study on linguistic descriptions by naive users of rules for defining smart home behaviors exploring the preferred and natural order of clauses containing events, states, or actions.

2.1.1. Methods

In the first qualitative study (Study1A; [57]), we explored how 11 Italian adult non-programmer participants articulated the definition of trigger-action rules by choosing among alternative conjunctions (“*if*”, “*when*”, “*while*”, “*as soon as*”, and “*as long as*”), verbal structures (e.g., “*it ranis*”, “*it is raining*”, and “*it starts to rain*”), and order of primitives (action-first such as “*close the window if it rains*”, and trigger-first “*if it rains close the window*”). We employed the thinking-aloud procedures [58, 59] to investigate participants’ mental models, and we video and audio recorded every session. The resulting verbal reports were analyzed using thematic content analysis [60].

In the second online experiment (Study1B), 64 participants Italian adult non-programmer participants evaluated pairs of trigger-action rules that differed only in the introductory temporal conjunction before the trigger (e.g., “*while <event>, <action>*” versus “*if <event>, <action>*”). The task was to choose the most accurate sentence in each pair for every scenario (adapted from [40]). We employed two stimuli categories specifically designed to support occurrences with different temporal features (state and events; [21]). The introductory conjunction of each rule could be specific for the occurrence (“*when*” and “*as soon as*” for events, and “*while*” and “*as long as*” for states), or generic (“*if*” which is suitable for both states and events). In addition, rule pairs could be presented either with a verbal structure specific per occurrence (e.g., “*it is raining*” for states, and “*it starts to rain*” for events) or with a generic one (e.g., “*it rains*”).

We designed the third online study to explore the event-state-action order naturally used by naive users (Study1C). We planned to recruit two groups of participants (expert programmers as the control group vs. non-programmers as the experimental group). Each participant will see 12 videos (one at a time) illustrating different home automation systems scenarios containing an event, a state, and an action. Participants will be asked to write the instructions in Italian to handle the scenario proposed for each video. After collecting the data, we will perform a linguistic analysis using an NLP tool (e.g., STANZA) on the descriptions written by participants investigating the order of elements and other linguistic features.

2.2. Objective 2: End-User Programming by Teachers

The second objective aims to develop a EUP interface based on specific linguistic aspects for guiding teachers in customizing their smart devices. Specifically, we want to investigate composition strategies, debugging, reasoning strategies, and mental models' assumptions during EUP tasks by teachers. We address two research questions:

RQ4. Which reasoning strategies and mental models are assumed for composing and debugging trigger action rules by teachers?

RQ5. Do some linguistic features implemented in an EUP interface influence the assumption of different reasoning strategies (specifically, abstract and concrete) and more effective mental models during EUP tasks by teachers?

To answer these questions, we planned three studies:

Study2A. A first qualitative pilot study with teachers exploring mental models and reasoning strategies assumption during trigger-action rules composition, debugging, and interpretation.

Study2B. A study with teachers investigating if linguistic primitives in the interface (abstract vs. concrete) affect the reasoning strategy assumed by teachers, their mental models, and their performance in composition and debugging trigger-action rules.

Study2C. A longitudinal study on EUP with teachers to study extended interaction in a real context.

2.2.1. Methods

First, we have designed and developed a system to support and promote math teachers in effectively using and programming a smart device with primary school children. The system includes a tangible IoT device and a EUP interface to personalize it (see Fig.1).

We developed a tangible educational IoT device named SMARTER (see [61] for the first prototype and Fig.1 for the evolved prototype[62]). Such a device was designed to support learning mathematics for children in primary school and to be flexible support for teachers in implementing simple interactive math games and tasks. Teachers could customize the tool's functioning through a set of language primitives for actions, states, and events designed and implemented as part of an existing interface named SENSATION [63, 64]. Primitives for actions consist of commands to control visual and acoustic feedback (e.g., "*Turn on blue LED SMARTER*"). States and events describe, respectively, the operations that teachers and children can perform on the tool (i.e., insertion and removal of tiles) and the tool configuration at that moment. An example of an event description is "*WHEN a digit tile is inserted*", while an example of a state is "*WHILE the position to the right of the inserted tile is empty*" (see Fig.1).

We have closed the first pilot study (Study2A) with five mathematics teachers to explore the strategies and mental models assumed during trigger-action rules composition [64] and debugging [65]. We designed Study2B to investigate trigger-action rules composition and debugging strategies by math teachers using two different versions of SENSATION for customizing SMARTER. One version was implemented using verbal primitives for guiding users in creating concrete rules (i.e., case-specific), and the other in creating abstract ones (i.e., generalizable). At the moment, we are finalizing the features of linguistic primitives for these two versions of SENSATION.

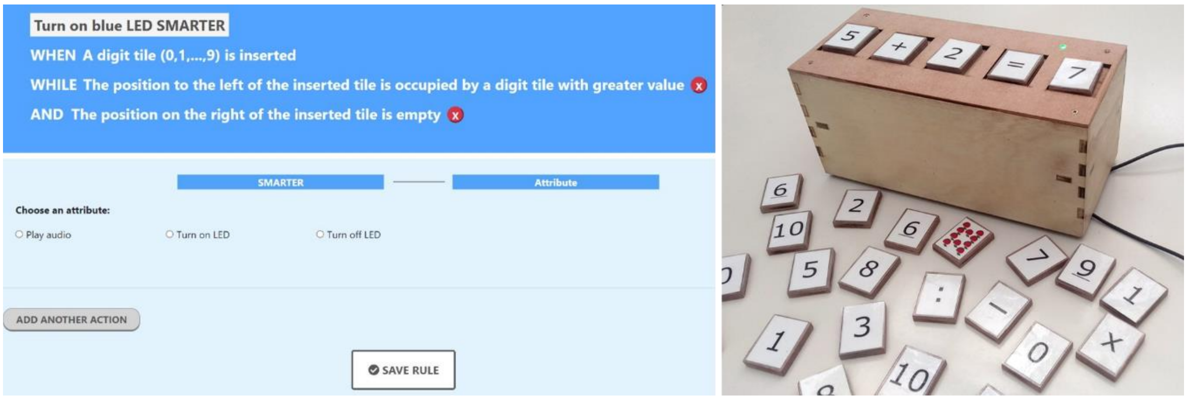

Figure 1: A screenshot of the authoring interface SENSATION (left side) and a picture of SMARTER evolved prototype (right side).

3. Conclusions

During the first year and a half of my Ph.D., I designed and conducted two studies (Study1A; Study1B) on language preferences in trigger-action rules by non-programmers. Preliminary results from qualitative (see [57]) and quantitative analyses showed that specific conjunctions (rather than the generic "*if*") are naturally preferred and clearly support the distinction between events with different temporal features, confirming the findings of other recent studies [27, 28, 63]. Even if the Italian linguistic aspects we investigated in our studies (i.e., conjunctions and verb structures) have similar semantics to those of other European languages (e.g., the trigger-action rules structure in Italian "*se-allora*" is comparable to the English form "*if-then*"), some possible subtle linguistic differences need to be further investigated.

In addition, I designed and participated in the development of an IoT device (SMARTER) and the related EUP interface. We employed the interface in a pilot study (Study2A) with mathematics teachers to explore the strategies and mental models assumed during trigger-action rules composition and debugging. Our qualitative results indicate that participants who reasoned about the states were facilitated in creating complete and comprehensive rules. Those who focused on actions tended to create overly simplified or redundant rules [64]. We identified three different debugging strategies implemented by participants (see [65]), partly similar to those already observed in novice programmers [66, 67, 68]. We just started Study1C to explore the order of elements naturally used by naive users, and we are going to test math teachers in the next few months (Study2B).

These studies, along with the others, will contribute to the development of a conceptual language framework for EUP considering the fundamental role of reasoning strategies and mental models assumed by naive users.

4. Acknowledgments

My Ph.D. project is supervised by Massimo Zancanaro and Barbara Treccani and funded by a grant from Fondazione Bruno Kessler (FBK). The work is partially supported by the Italian Ministry of University and Research (MIUR) under grant PRIN 2017 "EMPATHY: EMpowering People in deAling with internet of THings ecosYstems".

5. References

[1] M. Al-Emran, S. I. Malik, M. N. Al-Kabi, A Survey of Internet of Things (IoT) in Education: Opportunities and Challenges, in: Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications, in: A. E. Hassanien, R. Bhatnagar, N. E. M. Khalifa, M. H. N. Taha (Eds.), Studies in Computational Intelligence, vol. 846. Cham: Springer International Publishing, 2020, pp. 197–209. doi: 10.1007/978-3-030-24513-9_12

[2] F. Moreira, M. J. Ferreira, A. Cardoso, Higher Education Disruption Through IoT and Big Data: A Conceptual Approach, in: Learning and Collaboration Technologies. Novel Learning Ecosystems, in: P. Zaphiris e A. Ioannou (Eds.), Lecture Notes in Computer Science, vol. 10295. Cham: Springer International Publishing, 2017, pp. 389–405. doi: 10.1007/978-3-319-58509-3_31.

[3] D. D. Ramlowat, B. K. Pattanayak, Exploring the Internet of Things (IoT) in Education: A Review, in: Information Systems Design and Intelligent Applications, in: S. C. Satapathy, V. Bhateja, R. Somanah, X.-S. Yang, R. Senkerik (Eds.), Advances in Intelligent Systems and Computing, vol. 863. Singapore: Springer Singapore, 2019, pp. 245–255. doi: 10.1007/978-981-13-3338-5_23.

[4] R. Waranusast, A. Bangngoen, J. Thipakorn, Interactive tangible user interface for music learning, in: 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), IEEE, 2013, pp. 400–405.

[5] W. Almukadi, A. L. Stephane, BlackBlocks: tangible interactive system for children to learn 3-letter words and basic math, in: Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces, 2015, pp. 421–424.

[6] A. Girouard, E. T. Solovey, L. M. Hirshfield, S. Ecott, O. Shaer, R. J. Jacob, Smart Blocks: a tangible mathematical manipulative, in: Proceedings of the 1st international conference on Tangible and embedded interaction, 2007, pp. 183–186.

[7] J. Marquez, J. Villanueva, Z. Solarte, A. Garcia, IoT in Education: Integration of Objects with Virtual Academic Communities, in: New Advances in Information Systems and Technologies, in: Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, M. Mendonça Teixeira (Eds.), Advances in Intelligent Systems and Computing, vol. 444. Cham: Springer International Publishing, 2016, pp. 201–212. doi: 10.1007/978-3-319-31232-3_19.

[8] G. E. Forman, P. B. Pufall, Constructivism in the computer age. In: The Jean Piaget Symposium series, in: Hillsdale, N.J: L. Erlbaum (Eds.), University of Michigan, 1988.

[9] F. T. Moreira, A. M. B. Magalhães, F. M. S. Ramos, M. Vairinhos, The Power of the Internet of Things in Education: An Overview of Current Status and Potential, in: Citizen, Territory and Technologies: Smart Learning Contexts and Practices: Proceedings of the 2nd International Conference on Smart Learning Ecosystems and Regional Development, volume 2, Springer International Publishing, 2017, pp. 51-63.

[10] S. Benavides-Varela, C. Zandonella Callegher, B. Fagiolini, I. Leo, G. Altoè, D. Lucangeli, Effectiveness of digital-based interventions for children with mathematical learning difficulties: A meta-analysis, Computers & Education, vol. 157 (2020), p. 103953, doi: 10.1016/j.compedu.2020.103953.

[11] A. Delgado, L. Wardlow, K. O'Malley, K. McKnight, Educational Technology: A Review of the Integration, Resources, and Effectiveness of Technology in K-12 Classrooms, JITE:Research, vol. 14 (2015), pp. 397–416, doi: 10.28945/2298.

[12] S. Wiedenbeck, Facilitators and Inhibitors of End-User Development by Teachers in a School Environment, in: 2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05), Dallas, TX, USA: IEEE, 2005, pp. 215–222. doi: 10.1109/VLHCC.2005.36.

[13] S. Wiedenbeck, A. Engebretson, Comprehension Strategies of End-User Programmers in an EventDriven Application, in: 2004 IEEE Symposium on Visual Languages - Human Centric Computing, Rome: IEEE, 2004, pp. 207–214. doi: 10.1109/VLHCC.2004.12.

[14] H. Lieberman, F. Paternò, M. Klann, V. Wulf, End-User Development: An Emerging Paradigm», in: End User Development, in: H. Lieberman, F. Paternò, V. Wulf (Eds.), Human-Computer Interaction Series, vol. 9. Dordrecht: Springer Netherlands, 2006, pp. 1–8. doi: 10.1007/1-4020-5386-X_1.

[15] P. Markopoulos, J. Nichols, F. Paternò, V. Pipek, Editorial: End-User Development for the Internet of Things, ACM Trans. Comput.-Hum. Interact., vol. 24 (2017), pp. 1–3, doi: 10.1145/3054765.

[16] B. R. Barricelli, F. Cassano, D. Fogli, A. Piccinno, End-user development, end-user programming and end-user software engineering: A systematic mapping study, Journal of Systems and Software, vol. 149 (2019), pp. 101–137, doi: 10.1016/j.jss.2018.11.041.

[17] A. J. Ko, B. A. Myers, H. H. Aung, Six Learning Barriers in End-User Programming Systems, in: 2004 IEEE Symposium on Visual Languages - Human Centric Computing, Rome, Italy: IEEE, 2004, pp. 199–206. doi: 10.1109/VLHCC.2004.47.

[18] F. Paternò, C. Santoro, End-user development for personalizing applications, things, and robots, International Journal of Human-Computer Studies, vol. 131 (2019), pp. 120–130, doi: 10.1016/j.ijhcs.2019.06.002.

[19] A. Bellucci, A. Vianello, Y. Florack, L. Micallef, G. Jacucci, Augmenting objects at home through programmable sensor tokens: A design journey, International Journal of Human-Computer Studies, vol. 122 (2019), pp. 211–231, doi: 10.1016/j.ijhcs.2018.09.002.

[20] W. Brackenbury et al., How Users Interpret Bugs in Trigger-Action Programming, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow Scotland Uk: ACM, 2019, pp. 1–12. doi: 10.1145/3290605.3300782.

[21] J. Huang, M. Cakmak, Supporting mental model accuracy in trigger-action programming, in: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15, Osaka, Japan: ACM Press, 2015, pp. 215–225. doi: 10.1145/2750858.2805830.

[22] B. Ur, E. McManus, M. Pak Yong Ho, M. L. Littman, Practical trigger-action programming in the smart home, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Toronto Ontario Canada: ACM, 2014, pp. 803–812. doi: 10.1145/2556288.2557420.

[23] A. Rahmati, E. Fernandes, J. Jung, A. Prakash, IFTTT vs. Zapier: A Comparative Study of TriggerAction Programming Frameworks, arXiv (2017).

[24] A. K. Dey, T. Sohn, S. Streng, J. Kodama, iCAP: Interactive Prototyping of Context-Aware Applications, in: K. P. Fishkin, B. Schiele, P. Nixon, A. Quigley (Eds.), Pervasive Computing, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 254–271

[25] B. Ur et al., Trigger-Action Programming in the Wild: An Analysis of 200,000 IFTTT Recipes, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose California USA: ACM, 2016, pp. 3227–3231. doi: 10.1145/2858036.2858556.

[26] C. Ardito, P. Buono, G. Desolda, M. Matera, From smart objects to smart experiences: An enduser development approach, International Journal of Human-Computer Studies, vol. 114 (2018), pp. 51–68, doi: 10.1016/j.ijhcs.2017.12.002.

[27] G. Gallitto, B. Treccani, 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: G. Desolda, V. Deufemia, C. Gena, M. Matera, F. Paterno, B. Treccani (Eds.), CEUR Workshop Proc, CEUR-WS, 2020, pp. 7–11.

[28] M. Zancanaro, G. Gallitto, D. Yem, B. Treccani, Improving Mental Models in IoT End-User Development, Hum.-centric Comput. Inf. Sci., vol. 12 (2022), doi: 10.22967/HCIS.2022.12.048.

[29] C. W. Tan, A. Goh, Implementing ECA rules in an active database, Knowledge-Based Systems, vol. 12 (1999), pp. 137–144, doi: 10.1016/S0950-7051(99)00028-3.

[30] J. Bae, H. Bae, S.-H. Kang, Y. Kim, Automatic Control of Workflow Processes Using ECA Rules, IEEE Trans. on Knowl. and Data Eng., vol. 16 (2004), pp. 1010–1023, doi: 10.1109/TKDE.2004.20.

[31] J. J. Cañas, A. Antolí, J. F. Quesada, The role of working memory on measuring mental models of physical systems, Psicológica, 22 (2001), pp.25-42.

[32] P. N. Johnson-Laird, Mental models and human reasoning, Proceedings of the National Academy of Sciences, vol. 107 (2010) pp. 18243–18250, doi: 10.1073/pnas.1012933107.

[33] D. Norman, The design of everyday things: Revised and expanded edition, Basic books, 2013.

[34] S. Yarosh, P. Zave, Locked or Not?: Mental Models of IoT Feature Interaction, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver Colorado USA: ACM, 2017, pp. 2993–2997. doi: 10.1145/3025453.3025617.

[35] D. A. Norman, Some observations on mental models, in: Mental models, Psychology Press, 2014, pp. 15–22.

[36] T. Ngo, J. Kunkel, J. Ziegler, Exploring Mental Models for Transparent and Controllable Recommender Systems: A Qualitative Study, in: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, Genoa Italy: ACM, 2020, pp. 183–191. doi: 10.1145/3340631.3394841.

[37] T. Kulesza, S. Stumpf, M. Burnett, I. Kwan, Tell me more?: the effects of mental model soundness on personalizing an intelligent agent, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin Texas USA: ACM, 2012, pp. 1–10. doi:10.1145/2207676.2207678.
[38] F. Casati, S. Castano, M. Fugini, I. Mirbel, B. Pernici, Using patterns to design rules in workflows, IEEE Transactions on Software Engineering, vol. 26 (2000), p. 760.
[39] F. Pianesi, A. C. Varzi, Events and event talk, Speaking of events (2000), pp. 3–47.
[40] M. de Vega, M. Rinck, J. M. Díaz, I. León, Figure and Ground in Temporal Sentences: The Role of the Adverbs When and While, Discourse Processes, vol. 43 (2007), pp. 1–23, doi:10.1080/01638530709336891.
[41] N. C. Debnath, S. Banerjee, G. U. Van, P. T. Quang, D. N. Thanh, An ontology based approach towards end user development of iot, in: B. Gupta, A. Bandi, M. Hossain, (Eds.), EPIC Sre. Comp., EasyChair, 2022, pp. 1–10.
[42] T. Givón, The grammar of referential coherence as mental processing instructions, 30 (1992), pp. 5-56, https://doi.org/10.1515/ling.1992.30.1.5.
[43] J. F. Pane, C. A. Ratanamahatana, B. A. Myers, Studying the language and structure in non-programmers' solutions to programming problems, International Journal of Human-Computer Studies, vol. 54 (2001), pp. 237–264, doi: 10.1006/ijhc.2000.0410.
[44] M. Clark, P. Dutta, M. W. Newman, Towards a natural language programming interface for smart homes, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, Heidelberg Germany: ACM, 2016, pp. 49–52. doi: 10.1145/2968219.2971443.
[45] J. Good, K. Howland, Programming language, natural language? Supporting the diverse computational activities of novice programmers, Journal of Visual Languages & Computing, vol. 39 (2017), pp. 78–92, doi: 10.1016/j.jvlc.2016.10.008.
[46] A. F. Blackwell, First steps in programming: a rationale for attention investment models, in: Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments, Arlington, VA, USA: IEEE Comput. Soc, 2002, pp. 2–10. doi: 10.1109/HCC.2002.1046334.
[47] J.-M. Hoc, A. Nguyen-Xuan, Language semantics, mental models and analogy, Psychology of programming, Elsevier, 1990, pp. 139–156.
[48] J. Bonar, E. Soloway, Preprogramming knowledge: A major source of misconceptions in novice programmers, Human–Computer Interaction, vol. 1 (1985), pp. 133–161.
[49] H. Markovits, Conditional Reasoning, Representation, and Empirical Evidence on a Concrete Task, The Quarterly Journal of Experimental Psychology Section A, vol. 40 (1988), pp. 483–495, doi: 10.1080/02724988843000032.
[50] H. Markovits, H. Lortie-Forgues, Conditional Reasoning With False Premises Facilitates the Transition Between Familiar and Abstract Reasoning: Conditional Reasoning With False Premises, Child Development, vol. 82 (2011), pp. 646–660, doi: 10.1111/j.1467-8624.2010.01526.x.
[51] H. Markovits, R. Vachon, Conditional reasoning, representation, and level of abstraction, Developmental Psychology, vol. 26 (1990), pp. 942–951, doi: 10.1037/0012-1649.26.6.942.
[52] N. O. Ezeamuzie, J. S. Leung, Computational thinking through an empirical lens: A systematic review of literature, Journal of Educational Computing Research, vol. 60 (2022), pp. 481–511.
[53] S. Grover, R. Pea, Computational thinking in K–12: A review of the state of the field, Educational researcher, vol. 42 (2013), pp. 38–43.
[54] J. M. Wing, Computational thinking, Commun. ACM, vol. 49 (2006), pp. 33–35, doi:10.1145/1118178.1118215.
[55] B. R. Barricelli, D. Fogli, A. Locoro, EUDability: A new construct at the intersection of End-User Development and Computational Thinking, Journal of Systems and Software, vol. 195 (2023), p. 111516, doi: 10.1016/j.jss.2022.111516.
[56] K. M. Galotti, W. F. Ganong, What non-programmers know about programming: Natural language procedure specification, International Journal of Man-Machine Studies, vol. 22 (1985), pp. 1–10, doi: 10.1016/S0020-7373(85)80073-0.
[57] M. Andrao, B. Treccani, M. Zancanaro, Language and Temporal Aspects: A Qualitative Study on Trigger Interpretation in Trigger-Action Rules, IS-EUD, Cagliari, June 2023.

[58] T. Boren, J. Ramey, Thinking aloud: reconciling theory and practice, IEEE Trans. Profess. Commun., vol. 43 (2000), pp. 261–278, doi: 10.1109/47.867942.
[59] D. M. Russell, E. H. Chi, Looking Back: Retrospective Study Methods for HCI, in: J. S. Olson, W. A. Kellogg (Eds.), Ways of Knowing in HCI, New York, NY: Springer New York, 2014, pp.373–393. doi: 10.1007/978-1-4939-0378-8_15.
[60] V. Braun, V. Clarke, Using thematic analysis in psychology, Qualitative Research in Psychology, vol. 3 (2006), pp. 77–101, doi: 10.1191/1478088706qp063oa.
[61] M. Andrao, G. Desolda, F. Greco, R. Manfredi, B. Treccani, M. Zancanaro, SMARTER: an IoT learning game to teach math, in: Proceedings of the 2022 International Conference on Advanced Visual Interfaces, 2022, pp. 1–3.
[62] M. Andrao, Cappelletti, A., G. Desolda, F. Greco, B. Treccani, M. Zancanaro, SMARTER Than Before: Empowering Teachers to Program a Modular IoT Educational Device, in: Bellucci A., De Russis L., Diaz P., Morch A., Fogli D., Paternò F. (Eds.), Workshops, Work in Progress, Demos and Doctoral Consortium at IS-EUD 2023.
[63] G. Desolda, F. Greco, F. Guarnieri, N. Mariz, M. Zancanaro, SENSATION: An Authoring Tool to Support Event–State Paradigm in End-User Development, in: Human-Computer Interaction – INTERACT 2021, in: C. Ardito, R. Lanzilotti, A. Malizia, H. Petrie, A. Piccinno, G. Desolda, K. Inkpen (Eds.), Lecture Notes in Computer Science, vol. 12933. Cham: Springer International Publishing, 2021, pp. 373–382. doi: 10.1007/978-3-030-85616-8_22.
[64] M. Andrao, G. Desolda, F. Greco, R. Manfredi, B. Treccani, M. Zancanaro, End-User Programming and Math Teachers: an Initial Study, in: Proceedings of the 2022 International Conference on Advanced Visual Interfaces, 2022, pp. 1–3.
[65] R. Manfredi, M. Andrao, F. Greco, G. Desolda, B. Treccani, M. Zancanaro, Toward a Better Understanding of End-User Debugging Strategies: A Pilot Study, in: Fabrizio Balducci F., Breve B., Cena F., Mattioli A., e Rizvi M. (Eds.), CEUR Workshop Proc., CEUR-WS, 2022, pp. 31–35.
[66] S. Fitzgerald et al., Debugging: finding, fixing and flailing, a multi-institutional study of novice debuggers, Computer Science Education, vol. 18 (2008), pp. 93–116.
[67] S. Fitzgerald, R. McCauley, B. Hanks, L. Murphy, B. Simon, C. Zander, Debugging from the student perspective, IEEE Transactions on Education, vol. 53 (2009), pp. 390–396.
[68] C. Li, E. Chan, P. Denny, A. Luxton-Reilly, E. Tempero, Towards a framework for teaching debugging, in: Proceedings of the Twenty-First Australasian Computing Education Conference, 2019, pp. 79–86.