CEUR-WS.org/Vol-3408/short-s0-01.pdf

SMARTER Than Before: Empowering Teachers to Program a Modular IoT Educational Device

Margherita Andrao[1,2], Alessandro Cappelletti[2], Giuseppe Desolda[3], Francesco Greco[3], Barabara Treccani[1] and Massimo Zancanaro[1,2]

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

Abstract
Manipulation of tangible objects helps children in both learning and development, while digital technology creates a motivating and engaging environment for children. Tangible interactive tools can effectively combine these two aspects. Nonetheless, to exploit their potential as educational devices, it is necessary to empower teachers in customizing their behaviors. This paper presents the evolution of our first prototype: SMARTER 2.0, a tangible interactive device supporting teachers and children in different learning domains. We adapted the tool design and re-engineered the system architecture, aiming to enable teachers to define learning activities and subject fields more expressively and flexibly. A fundamental part of our work is to allow teachers to define games, modality, and application of SMARTER through its integration with an End-User Development environment.

Keywords
End-User Development, Internet of Things, Educational Technologies

1. Introduction

In education, tangible tools have a recognized potential for improving and supporting various aspects of learning. The physical interaction with the world (i.e, the grasping and manipulation of concrete, physical objects) allows children to concretely explore abstract concepts, which supports their cognitive development and learning [1]. In recent decades, a great effort focused on creating and studying the application of tangible smart devices in the educational field to support the learning-teaching process and exploits its potential [2, 3, 4]. For example, tangible and interactive devices were designed for supporting music [5], math and language learning for young children (e.g., [6, 7]). Using smart devices creates an interactive and engaging learning environment (e.g., [2, 8]). They may enable students to construct their own knowledge [9], while teachers can receive real-time data and information about students' learning achievements, performance, and errors [2, 3, 10]. All these elements can facilitate the customization of

IS-EUD 2023: 9th International Symposium on End-User Development, 6-8 June 2023, Cagliari, Italy
margherita.andrao@unitn.it (M. Andrao); cappelle@fbk.eu (A. Cappelletti); giuseppe.desolda@uniba.it (G. Desolda); francesco.greco@uniba.it (F. Greco); barbara.treccani@unitn.it (B. Treccani); massimo.zancanaro@unitn.it (M. Zancanaro)
0000-0003-2245-9835 (M. Andrao); 0000-0002-2764-3132 (A. Cappelletti); 0000-0001-9894-2116 (G. Desolda); 0000-0003-2730-7697 (F. Greco); 0000-0001-8028-0708 (B. Treccani); 0000-0002-1554-5703 (M. Zancanaro)
© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

curricula to accommodate individual student abilities, which is essential to the learning process, particularly for those with specific learning disabilities [11]. Some recent works focused on creating tangible interaction devices for supporting individuals with specific learning disabilities, such as dyslexia [12, 13, 14] and dyscalculia [15]. Within this research area, we are investigating smart devices for supporting teaching to primary school children. Specifically, we recently designed and developed SMARTER (SMART E-Rods [16]), a tangible interactive tool based on Internet of Things technology for math, inspired by the non-digital "Cusinarie rods" [17, 18], also designed as a possible dyscalculia training device. The underlying idea was to support teachers in teaching mathematical concepts (specifically, comparing quantities and arithmetic operations) by providing a tangible, interactive, and engaging environment suitable for children. Teachers can customize SMARTER behavior (and specifically, its visual and audio feedback on children's actions) according to their educational goals and the children's needs, ages, and skills through SENSATION, an End-User Programming platform for visual trigger-action programming [19, 20].

Since teachers are the primary experts in the educational field, our goal has expanded and evolved in giving them more freedom/control and expressiveness in designing their own games and application contexts on the device. For this purpose, we adapted the tool design and architecture, aiming to create a more flexible system in which teachers can define the games' structures and their applications in different subjects (math, language, music, etc.). This demo presents SMARTER 2.0, the evolution of our first prototype, a modular tangible educational tool that can be integrated with EUD environments.

2. SMARTER 2.0

The version 2.0 of SMARTER aims to increase the flexibility of its application contexts and empower teachers to define its behavior. The new architecture of SMARTER has been designed to provide a modular system with as little coupling as possible between modules (see Figure 1). The following subsections describe the structural changes and re-engineering process we have performed to achieve this goal.

2.1. SMARTER device

Originally, SMARTER [16] consisted of (i) a cardboard box with a surface of 29.7 x 21 cm in which the game area is printed (5 slots corresponding to the 5 RFID readers attached under the surface) and containing a speaker and an RGB LED, respectively for audio and visual feedback; (ii) a set of tangible tiles (with RFID tags attached) representing rods (for the quantity from 1 to 10), math operators and symbols. The five slots in the game area had different sizes and shapes to guide children in composing the games by placing in the correct slots the corresponding class of tiles (e.g., rod tiles should be placed in the 3rd and 5th slots in comparing quantities game).

From a technological point of view, the structure of SMARTER 2.0 is identical to the previous version: a NodeMCU ESP8266 board connected with 5 RFID readers (each one representing a slot for placing tiles), a speaker, and an RGB LED providing audio and visual feedback, respectively.

The most important update regards the design of the form factor: the current version is a plywood box with the surface resized to 26 x 11 cm (see Figure 2). We made this design choice by

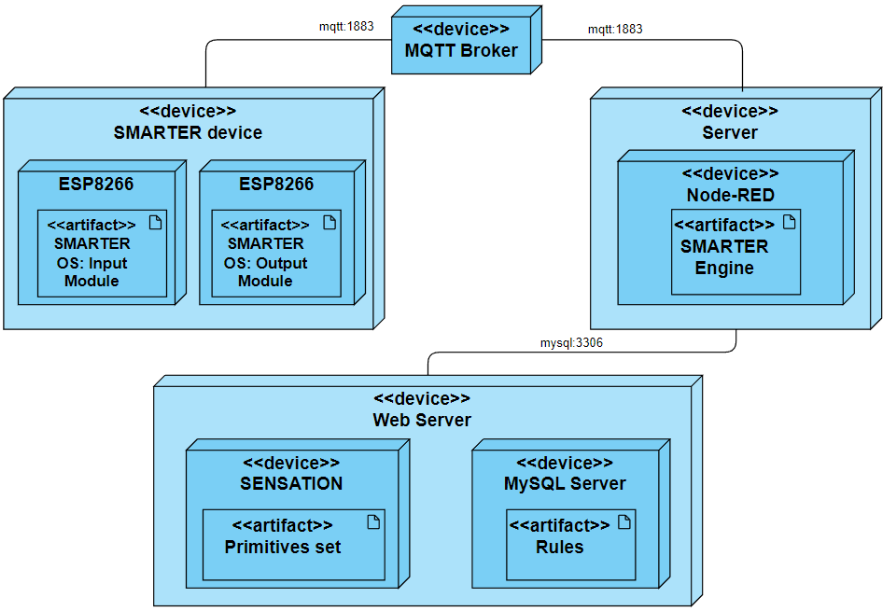

Figure 1: Deployment Diagram for the architecture of SMARTER 2.0.

considering the real classroom context, i.e., the size of school desks in primary school, assuming a possible future scenario where, for each child (in a single desk), several SMARTER boxes can be joined together (horizontally or vertically) to create a larger box to accommodate designs for more complex games. In addition, the five slots now have the same shape and dimension. Although children are no longer intuitively guided to place certain classes of tiles in defined slots (as in the first version), removing this constraint has allowed us to increase the system flexibility, the range of games, and contexts/subjects in which SMARTER can be employed. For example, SMARTER 2.0 is suitable for supporting arithmetic operations, single- or multi-digit quantities comparisons, and sorting numbers in ascending or descending order.

Another improvement concerns the tiles, which represent different objects of the chosen domain (in our current case, they are math symbols but they could possibly be letters, musical notes, etc). The set of tiles in this version (in which Mifare® ultralight RFID tags are embedded) represent digits and quantities (from 0 to 9), operators (+, -, ×, ÷), symbols (=, <, >). See Figure 2.

2.2. SMARTER OS

SMARTER 2.0 runs an ad-hoc SMARTER Operating System (OS), which allows the communication and retrieval of basic information and the actuation of basic actions using the MQTT protocol. In particular, the *actions* consist of turning on/off the device LED, playing a sound, stopping the speaker, and resetting the board. Moreover, the OS allows the device to communicate basic *events*, namely the insertion and removal of a tile. Finally, the values placed on the individual readers at any given time can be queried, as well as the entire configuration

Figure 2: The figure presents two pictures, one representing one SMARTER 2.0 and math tiles (left side) and the other a possible scenario with two SMARTER modules joined horizontally (right side).

of the board (5 values, one for each reader): the latter represents the *state* of the board. The formalization of the MQTT commands is reported in Table 1 - Appendix A.

The SMARTER OS is written in C++ and is split into two modules, to be loaded separately onto two ESP8266 boards, which compose a SMARTER device. Both modules must be set with the same device_ID. The first one is the input module which manages the functioning of the readers and the handling of MQTT messages for the *events* and *states*. Therefore, it registers events on the RFID readers and communicates them to the MQTT broker as soon as they happen. It also listens for state requests and communicates them to the broker, either the full configuration or a single reader, according to the incoming request. The second one is the output module which manages the actuators (the LED light and the speaker), i.e., the *actions*. Therefore, it listens for messages containing commands and performs the according action. The tiles are loaded with any string value of 255 bytes. The semantics of the tile (e.g., if it represents a digit, a symbol, etc.) is not interpreted by the OS, but by the SMARTER Engine (Subsection 2.3). The tiles can be written with any RFID writer compatible with Mifare®.

The implementation of an operating system to manage the basic operations of the device allows it to work stand-alone, regardless of the integration with other software modules. In fact, a SMARTER device could be even controlled remotely by directly sending and receiving handcrafted MQTT messages that follow the formalism defined in Table 1 - Appendix A. This allows easier debugging and testing, since the functioning is completely encapsulated and uncoupled from the games' logic (independently of the tiles representing numbers, letters, etc.). Moreover, complex behavior could be simulated by a human agent in a Wizard-of-Oz fashion.

2.3. SMARTER Engine

A SMARTER Engine is the core of the game's logic. It is the module that defines the semantics and the high-level behavior of rules and interprets the values of *tiles*. A *rule* is defined as a single event that, if verified, triggers one or more actions; a rule also contains zero or more states logically disjunct or conjunct (i.e., linked together by OR or AND statements). A rule also has a unique identifier. A *game* is defined by a set of one or more rules and works with a specific *tile set*. A *tile*, which carries a string value, is interpreted according to the tile set used by the engine. For example, the tile set "Arithmetic" may contain tiles such as digits (from 0 to 9), symbols (=, <, >), and operators (+, -, ×, ÷); these values are parsed starting from the raw string values contained in a tile which represent the corresponding digit, symbol or operator.

Each engine defines a set of primitives (actions, events, and states), which can be arbitrarily complex and/or domain-specific. For example, an engine can define the logic for the event *"A digit greater than 5 is placed"*, or for the state *"The board is empty"*. The engine reads and interprets rules that are defined in a specific JSON format. Rules can be manually written or be produced as the output of an EUD interface like SENSATION (see subsection 2.4), thus working with human-readable rules (referred to as "Rules" in Figure 1). In this way, it is possible to define distinct engines that implement different logics in different domains. Clearly, to be compatible with the system, an EUD tool must produce JSON rules that only contain primitives that are defined in the engine. A SMARTER Engine manages an MQTT channel to communicate with one or more SMARTER devices, receiving basic events, requiring states, and executing actions. Each SMARTER device can connect to at most one SMARTER Engine. This architecture allows the developer to create an engine that manages two or more SMARTER devices as if they were a single board, notably increasing the available positions for defining games using up to 10, 15, or more tiles. Any server can be used to implement the engine, as long as it can function as an MQTT client. Currently, we use Node-RED, a visual task automation tool that easily allows communication over MQTT and the definition of JavaScript-written submodules to implement a specific logic (as visible under "Server" in Figure 1).

2.4. End-User Development: SENSATION

The functionalities of SMARTER can be customized through SENSATION [19], an End-User Development platform that supports trigger-action programming. We have implemented language primitives in SENSATION (currently only for math games) to enable teachers themselves to define new games (see Figure 2.4). Specifically, they can compose extended Trigger-Action rules in the form: "DO action(s) WHEN event WHILE state(s)", cf. [21, 22]. 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"*.

SENSATION allows the software developer to define custom primitives sets (for actions, events, and states) to reflect the logic inside a SMARTER engine ("Primitives sets" in Figure 1). Obviously, the same primitives must be already defined within the SMARTER engine. SENSATION is written in Laravel and Vue.js and runs on an Apache web server. User-defined rules are saved in a MySQL database in a JSON format compatible with a SMARTER Engine; the latter can access these rules via MySQL Client/Server protocol and interpret them.

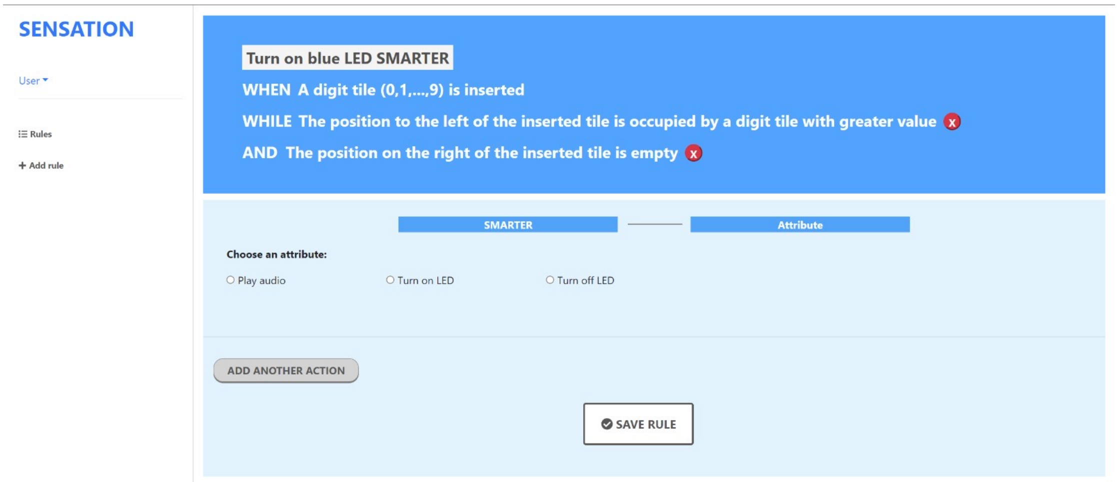

Figure 3: The figure presents a screenshot of the UI of SENSATION.

3. Conclusions and Future Work

In this demo, we presented SMARTER 2.0, which embodies our goals of creating a more flexible system. To achieve this purpose we applied modifications at design level and re-engineered the architecture. Currently, SMARTER 2.0 only supports the creation of math games, but we are working on the production of an engine for using this device in other domains, such as language (using tiles from a to z) and music (using tiles with notes). Also, we aim to fully leverage the system modularity by combining multiple SMARTER modules, allowing more complex and expressive game solutions depending on the teachers' goals (e.g., column operations, mathematical expressions, or words that need more than 5 slots). The value of our work, even imagining possible future scenarios, is the possibility to empower teachers in choosing the game domain, how many modules to use, creating new games, and deciding the feedback. During the demonstration, we will allow participants to create rules using SENSATION and observe their behavior directly on a version of SMARTER 2.0.

Acknowledgments

This 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". The research of Francesco Greco is funded by a PhD fellowship within the framework of the Italian "D.M. n. 352, April 9, 2022"- under the National Recovery and Resilience Plan, Mission 4, Component 2, Investment 3.3 - PhD Project "Investigating XAI techniques to help user defend from phishing attacks", co-supported by "Auriga S.p.A." (CUP H91I22000410007).

References

[1] J. Piaget, The early growth of logic in the child: Classification and seriation, volume 83, Routledge, 2013.

[2] M. Al-Emran, S. I. Malik, M. N. Al-Kabi, A survey of internet of things (iot) in education: Opportunities and challenges, Toward social internet of things (SIoT): Enabling technologies, architectures and applications: Emerging technologies for connected and smart social objects (2020) 197–209.

[3] 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: 4th International Conference, LCT 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part I 4, Springer, 2017, pp. 389–405.

[4] D. D. Ramlowat, B. K. Pattanayak, Exploring the internet of things (iot) in education: a review, in: Information Systems Design and Intelligent Applications: Proceedings of Fifth International Conference INDIA 2018 Volume 2, Springer, 2019, pp. 245–255.

[5] R. Waranusast, A. Bang-ngoen, 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.

[6] 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.

[7] 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.

[8] 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, Springer International Publishing, 2016, pp. 201–212.

[9] G. Forman, P. B. Pufall, Constructivism in the computer age, Psychology Press, 2013.

[10] F. T. Moreira, A. Magalhaes, F. 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-University of Aveiro, Portugal, 22-23, June 2017 2, Springer, 2018, pp. 51–63.

[11] S. Benavides-Varela, C. Z. 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 157 (2020) 103953.

[12] M. Fan, A. N. Antle, E. S. Cramer, Design rationale: opportunities and recommendations for tangible reading systems for children, in: Proceedings of the the 15th international conference on interaction design and children, 2016, pp. 101–112.

[13] M. Fan, A. N. Antle, M. Hoskyn, C. Neustaedter, E. S. Cramer, Why tangibility matters: A design case study of at-risk children learning to read and spell, in: Proceedings of the 2017 CHI conference on human factors in computing systems, 2017, pp. 1805–1816.

[14] S. Pandey, S. Srivastava, Tiblo: a tangible learning aid for children with dyslexia, in:

Procedings of the Second Conference on Creativity and Innovation in Design, 2011, pp. 211–220.
[15] G. Erfurt, E. Hornecker, J. Ehlers, S. Plaschkies, Hands-on math: A training system for children with dyscalculia, in: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, pp. 1–6.
[16] 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.
[17] W. H. Lucow, An experiment with the cuisenaire method in grade three, American Educational Research Journal 1 (1964) 159–167.
[18] R. A. Abreu-Mendoza, L. Coulanges, K. Ali, A. B. Powell, M. Rosenberg-Lee, From non-symbolic to symbolic proportions and back: a cuisenaire rod proportional reasoning intervention enhances continuous proportional reasoning skills, Frontiers in Psychology 12 (2021) 633077.
[19] 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: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part II 18, Springer, 2021, pp. 373–382.
[20] 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.
[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, 2015, pp. 215–225.
[22] M. Zancanaro, G. Gallitto, D. Yem, B. Treccani, Improving mental models in iot end-user development, HUMAN-CENTRIC COMPUTING AND INFORMATION SCIENCES 12 (2022).

A. Appendix: SMARTER OS MQTT primitives

Table 1
MQTT messages to interact with the SMARTER OS. Messages in black are commands to send to the SMARTER device, while messages in red are sent by the SMARTER OS to the MQTT broker.

Type	Topic	Payload	Description
Action	<device_ID>/action	action: "play", value:<file_name>	Play the audio file called <file_name>
Action	<device_ID>/action	action: "stop"	Stops audio
Action	<device_ID>/action	action: "led", value: <color>	Turn LED on with the color specified in <color>in RGB values (R,G,B)
Action	<device_ID>/action	action: "led", value:""	Turn LED off
Action	<device_ID>	"reset"	Reboots the device
Event	<device_ID>/event	event: "card_placed", reader: <i>, value: <x>	A tile with value <x>has been placed in position <i>
Event	<device_ID>/event	event: "card_removed", reader: <i>, value: <x>	A tile with value <x>has been removed from position <i>
State	<device_ID>	"state"	Requires the full board state
	<device_ID>/state	[5]{reader : <i>, value: <x>}	Returns the full board state (5 reader/value objects)
State	<device_ID>/<i>	"state"	Requires the state of the reader <i>
	<device_ID>/state	reader : <i>, value: <x>	Returns the value <x>present on the reader <i>("" if no tile is present)