View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by AIS Electronic Library (AISeL)

Proceedings of the 52nd Hawaii International Conference on System Sciences | 2019

Non-personal Data Collection for Toy User Interfaces

Anna Priscilla de Albuquerque
Universidade Federal de Pernambuco
apa@cin.ufpe.br

Judith Kelner
Universidade Federal de Pernambuco
jk@cin.ufpe.br

Abstract

Toy-user-interfaces (ToyUI) are computing devices or peripherals that leverage interactivity and connectivity with other devices to promote physical and social play. ToyUI products may collect both personal and non-personal data (NPD) on their users. We propose nine data patterns for NPD collection as part of ToyUI design based on the study of 297 ToyUI items from both the literature and industry. In addition, we introduce a printed circuit board (PCB) used for rapid prototyping that enabled NPD data collection concerning both objects and users by gathering non-personal identification, positioning system, and motion tracking. We demonstrate the effectiveness of our hardware architecture by embedding it into two design scenarios, namely, closed rules and open-ended rules solutions. The objectives here are to assist the ToyUI makers in creating more meaningful play experiences while ensuring the privacy of children's and their parents' data.

1. Introduction

Toys and game industries have significantly invested in what is known as "smart toys." These play products allow connectivity with other toys and devices such as smartphones, tablets, and game consoles. They fall into two classes, computing devices, and peripherals and usually combine physical computing with mobile services and augmented reality (AR) support [1]. We refer in this work to these play products as Toy-User-Interfaces (ToyUI). We can understand the concept of a ToyUI as a playful physical computing component (device or peripheral) that offers interactivity and connectivity in order to support leisure and social play activities for the users.

A ToyUI uses different computing technologies to obtain real-time data that it uses then to regulate play rules (e.g., tracking of both objects and users), and to provide contextual feedback to the users. Hence, ToyUI solutions may collect both non-personal data (NPD) and personal data (PD) related to their users. What differentiates PD from NPD collection is that PD collection contains directly or non-directly compromising information that may expose user privacy and allow the singling out of individual behaviors (e.g., voice recordings and facial pictures of the users). The ToyUI industry faces considerable challenges in an attempt to ensure the security and privacy of such data. Understandably, this industry would benefit from investing in products that limit themselves to exploring more the nonintrusive NPD to attract consumers.

A toy is a play product designed for promoting leisure, entertainment, and imaginative play experiences. With the miniaturization and lower costs of processing circuits, toys have become capable of collecting and processing data in real-time. However, the design of meaningful play experiences that are limited to NPD collection remains a challenging task for most toy makers. Thus, researchers are invited to fill this gap by addressing best design practices and providing adequate tools for the ToyUI community. In this paper, we aim to introduce a design approach for ToyUI projects that will focus on the NPD collection planning. The goals here are to provide both guidance and practical tools for toy makers and to assist them in creating more meaningful play experiences supported by efficient NPD collection. Considering that it is possible to collect NPD from both objects and the users, the NPD collection for ToyUI should address the following requirements:

- Non-personal identification
- Unidentifiable positioning system (e.g., 2D or 3D position, orientation, and relative position)
- Motion tracking (e.g., getting speed, acceleration, and angular frequency)

In Section 2 we discuss related work and the benefits of the NPD collection planning approach. In Section 3 we briefly describe the literature and industrial mapping, which resulted in a list of 297 ToyUI items. Then, Section 4 details the use of NPD collection for ToyUI by summarizing the types of

URI: https://hdl.handle.net/10125/59612
ISBN: 978-0-9981331-2-6
(CC BY-NC-ND 4.0)

collected data and by relating them to different technologies and design solutions. In Section 5 we discuss how data collection could be used to support different play rules (e.g., open-ended and closed rules), and we propose a list of nine data patterns for planning NPD collection. Next, in Section 6 we develop a new printed-circuit board (PCB) for rapid prototyping based on the elicited NPD collection requirements. Moreover, we demonstrate how to apply the PCB and the list of data patterns to plan two design solutions under both open-ended and closed play rules. Finally, we state our conclusions in Section 7, and we point out some improvements to be made in the hardware architecture as part of future works

2. Related work

The ToyUI research community is highly concerned with the support of security and encryption of the data collection process. This is reflected by the presence of guidelines for efficient privacy and access control policies [1, 2]. Currently, there are specific regulations for collecting underage user's data, such as the American *FTC Children's Online Privacy Protection Act* (COPPA), which several toy makers are using to define their privacy policies. For instance, Rafferty et al. [2] defined privacy requirements for guardians to have control over their child's data. These included the rights to request restriction of the data usage, to access and inspect the stored data, to request data deletion, and to be notified about any data sharing.

A Service Oriented Architecture (SOA) approach was adapted for ToyUI [1]. It focused on establishing new privacy-related layers (i.e., policy, security, and access control layers) to circumvent data disclosure and other privacy threats. This SOA architecture consisted of a service provider, a service requester, and a service broker, and supported three operations: publish, find, and bind. The publish layer used Universal Description Discovery Integration (UDDI), an OASIS standard for service providers. Then, the find and bind layers used two XML-based protocols, namely, the Web Services Description Language (WSDL) for managing the messages, and the Simple Object Access Protocol (SOAP) for conveying request and response messages. Design alternatives of a ToyUI architecture may explore the Message Queue Telemetry Transport (MQTT) from IBM, another service broker protocol mainly used in the implementation of Internet of Things (IoT) services. MQTT is a lightweight, simple and open protocol that is suitable for IoT applications since it is ideal for constrained environments and small bandwidth networks with limited processing capabilities [3].

Over the recent years, the ToyUI community has invested in both rapid prototyping and licensing technologies. Schmitz et al. [4] introduced flexible deformable materials allowing different conductive identification patterns on the mobile device's screen. Similarly, *Volumique* distributed a technology licensing to support conductive touch points for ToyUI figurines. Another licensing technology is the French *EPaw*, which uses a Near-field Communication (NFC) sensing game board able to connect with mobile devices. Such a game board can identify objects and their relative position on the board; hence, it acts as a physical extension of the mobile device's screen.

Similar to our work, Soute et al. [5] introduced *RaPIDO,* a toy-device for rapid prototyping of open-ended play games that explored NPD collection by using a Local Positioning System (LPS) and relative positioning tracking. Differently, we propose a more generic prototyping tool as a means to allow embedding our solution into multiple "toy-shells" of different shapes (e.g., a stuffed toy or a plastic sword). Some benefits of such a generic tool include an easy to use setup, which is suitable for several play rules and purposes. Furthermore, it may support the interoperability of different communication protocols in the same design (e.g., a setup supporting both Wi-Fi connectivity with a mobile device and NFC connectivity with distributed tags).

We acknowledge the importance to define these different architectural approaches and universal protocols to manage data sharing, behavior, and storage. Hence, our research results in a list of nine data patterns that incorporate these three features (i.e., sharing, behavior, and storage), as a means to support data collection planning. While focusing on the planning phase as a strategy to circumvent data disclosure and other security threats, we first ensure that data collection would not incorporate PD or any sensitive information. We seek to stimulate toy makers to limit their solution designs to a safer and efficient NPD collection planning. Moreover, we establish technical requirements to propose a rapid prototyping tool for the ToyUI community to support the NPD collection strategy.

3. ToyUI review

To investigate data collection for ToyUI, we consulted a list of 297 items among play products and literature prototypes. These ToyUI items were selected through a combined systematic literature mapping and an industrial review, which are available for consulting in the master's thesis [6]. Due to limited space for this

paper, we will briefly describe the two review procedures.

The systematic literature mapping was performed following the Kitchenham & Charters guidelines [7] and it covered publications from 2008 to 2016. Two researchers selected 102 items from 438 peer-reviewed papers in three selective phases. The mapping used automated search strings in the *ACM Library, IEEE Xplore, Springer Link, Scopus* and *Science Direct*, and complementary search sources (e.g., proceedings of international conferences and lists of bibliography references therein). The publications were selected using inclusion and exclusion criteria items, and quality assessment. The industrial report used multiple search sources (e.g., literature citations, toy brand's websites, and string results of the *Google's* site), and it selected 115 ToyUI products released from 2012 to 2016.

The research goal of this combined mapping was to identify the play features and interface features of these ToyUI items, as a means to understand how the smart toys currently support social and physical play user interfaces of different play setups. Thus, it investigated the types of toys and their materials, the networking technologies, peripherals and devices, including the physical and social play dynamics promoted by such toys, and the associated play contents and thematic. For this work, we updated the mapping by following the instructions provided in the master's thesis [6], which included 16 publications and 29 products releases since 2017. The research prototypes were then gathered from the 118 selected publications (2008—2017). Thus, the full list contained 297 ToyUI items (i.e., 144 products and 153 prototypes). An overview of the list contains:

- 30 connected action figurines for mobile applications or game consoles (e.g., Nintendo's Amiibo and Mattel's Apptivity)
- 24 wireless controlled toys (e.g., cars and toy-drones), 19 toy-robots, 9 smart building block sets, 9 AR-toys (e.g., AR-guns and fishing rods), 16 talking dolls and other 7 social toys
- 17 smart toys for physical exertion (e.g., smart balls and Frisbees), and 15 children´s wearables and gadgets (e.g., smartwatches, fashion accessories, and handheld devices)
- 18 (AR and mobile) board-games, 16 hybrid arcade games (e.g., smart floors and light shooting-guns), and 4 toys for pervasive gaming
- 106 "serious" toys (e.g., 36 educational toys, 26 programming toys, 30 therapeutic toys, and 14 toys for playful training)

These ToyUI items explored a range of computing technologies including AR-markers (e.g., fiducial markers and QR codes), depth sensors (e.g., *Microsoft's Kinect* and infrared (IR) tabletops), ultrasonic sensors and other embedded sensors (e.g., accelerometers, gyroscopes, and magnetometers). They used wireless communication protocols such as Wi-Fi, Bluetooth, Bluetooth Low Energy (BLE), ZigBee, XBee, NFC, and Radio-frequency Identification (RFID). Moreover, a range of devices and peripherals were present, including, personal computers, notebooks, mobile phones, tablets, game controllers (e.g., the *Nintendo's Wiimote*), digital cameras, microphones, speakers, head-mounted displays (HMD), and all size of displays (e.g., LEDs, projectors, TV, and computer monitors).

3.1. Data collection for ToyUI

To analyze the data collection techniques performed by of the 297 ToyUI items we consulted the contextual summaries of each research paper and the full-texts (when necessary) to find any unmapped information. Similarly, to extract data related to the industrial products, we consulted the available text, picture, and audio-visual materials from several web sources, (e.g., product's websites, professional blogs, newspapers, commercial videos, and consumer's reviews on the streaming platforms, like *YouTube* and *Vimeo*).

Summarizing the data collection of the 297 items, we identified that 55 of them collected PD from the users whereas another 170 items explored only NPD collection. The remaining 50 cases allegedly collected both NPD and PD, and we identified 22 items as being "sensitive to collect PD" since they used computing technologies (e.g., digital cameras and microphones) that may collect undesired PD without explicitly notifying the users. For example, a picture or a video intended to capture a printed AR-marker may accidently show the face of the user holding the marker, in the background. Another recurrent case was the presence of an embedded microphone. While it was destined to record the environment's sounds it might still, include the user's voice.

Overall, most of the PD collected by the ToyUI items were multimedia data. As personal multimedia data, we considered any text, image, video, or sound files that explicitly or implicitly provided information about an identifiable individual. For example, the full name of the individuals in their user's accounts, their face pictures or videos, and voice recordings. Other collected PD included the geo-positioning system (GPS) information, since GPS coordinates give means to estimate the real-time user's location and his/her geo-location preferences (e.g., the user's home, work address and recurrent locations).

The goals behind the ToyUI items to store PD are several. Pervasive location-based applications explore the GPS data to allow the play activities. Serious toys for education and therapy can store PD of the students and patients for further analysis of their performance by the educators and therapists, respectively. Both research prototypes and products may collect user's play data as means to improve their designs for future versions, or adding new play contents. Social toys, for example, are intended to promote communication among the users or between the user and the toy agent. In that sense, collecting PD including voice, image and video are standard practices for such toys.

However, there are social toys that promote co-located communication, which may not require the gathering of PD. Melonio and Rizvi [8] introduced the TurnTalk, a toy that assists groups of children to achieve a balanced conversation using NPD collection. The toy can count the turns that each participant took when holding a conversation. Each child uses an RFID card to start it talking turn, and at each turn, the toy lights up a LED bar. These LED bars provide visual information to the group so that they can be aware of their talking turns; then, the toy rewards the participants with plastic coins when they achieve a balanced conversation.

In the next section, we detail our approach NPD collection for ToyUI since this is the main contribution of the present research.

4. NPD collection for ToyUI

ToyUI collects NPD from both objects and the users. We identify three types of data that are suitable for both object and user's tracking; these are the non-personal identification (NPID), positioning system, and motion tracking. However, there are specific NPD which can only be collected by either an object or a user, these are:

- Non-personal multimedia for object's tracking (e.g., text, image, video, and audio)
- Biofeedback for user's tracking (e.g., heart rate, breath, brain waves, and electro-dermal information)

The ToyUI items can use these two types of NPD without directly identifying an individual. On the one hand, to collect non-personal multimedia data from the objects may demand the use of digital cameras or microphones, which can turn the ToyUI sensitive to collecting undesired PD. On the other hand, the biosensors are specifically designed sensors for gathering data from the living beings. The *ChillFish* [9], for example, is a therapeutic toy that uses a breath sensor data to allow the user to control the game inputs. In this work, we will focus on the types of NPD collection suitable for both object and user tracking. This will allow us to define a rapid prototyping tool for ToyUI that can support both design requirements.

4.1. Non-personal identification (NPID)

NPID consists of identifying either a real or virtual entity (object or user) without recognizing the entity as an identifiable individual. In other words, the ToyUI may identify the object's or user's ID as "player one" without assigning who the "player" is as an individual (e.g., a wristband's ID is "player one," so any user that wears it will be identified as the "player one"). NPID is an essential component for ToyUI since it supports the identification of objects and users to regulate the play rules. This can be achieved in different ways. Next, we list the different types of NPID and their meanings, based on what we found in the ToyUI mapping:

- **Single ID** represents a single object or user that is equal to a single value (e.g., a user's ID is equal to "player one")
- **Multiple ID** represents a single object or user that may represent multiple values (e.g., the six faces of a cube associate six IDs to a single object).
- **Collective ID** is a combination of two or more objects (or users) that creates a collective value (e.g., a collection of checkers' pieces may create a "player" ID, and two or more users wearing similar belts may create a "team" ID).
- **State ID** occurs when the object or user's ID may change its value to a different ID or exchange its ID with another ID (e.g., the object's ID was "red," and it turned to "blue" after another "blue" object has been detected).

Different computing technologies allow collecting NPID from both objects and users, and a typical approach is to use individual tags, such as RFID/NFC, IR-tags, and AR-markers (e.g., fiducial markers and QR codes). Another setup is to use computer vision techniques for feature detection and description (e.g., the OpenCV library) that allows recognizing the object's NPID by either using its shape, color, lighting, saturation, texture or other image descriptors.

Care must be taken when dealing with ToyUI setups that use AR-markers or feature detection approaches. They should avoid gathering any undesired PD. Solutions to this problem include positioning the digital cameras to create a limited detection field, as is the case in the Portico platform [10]. This platform uses two digital cameras placed in the corners of the tablet. This creates a detection field that holds the tablet's surface and a limited outside area. It allows detecting the objects placed upon the tablet's screen and around the device. A similar setup

appeared in the product Osmo that uses a mirrored device attached to the iPad's digital camera, which creates a detection field located in front of the device. In both cases, the collected images may include the user's hands when manipulating the objects, but the ToyUI can limit it to collect only NPD.

4.2. Non-personal positioning system

A positioning system is a mechanism that determines the location of a real or virtual entity in a designated space. The positioning system data may vary according to its accuracy and the degrees of freedom. In addition to GPS, there are other types of positioning systems that allow both object and user tracking, without collecting PD. These are:

- **2D positioning coordinates (X, Y)** can estimate the location of an object or user (his/her full body or body's parts) in a two-dimensional plane, (e.g., estimate the location of the object placed upon a table, and estimate the location of the user's body on the floor)
- **3D positioning coordinates (X, Y, Z)** can estimate the location of an object or user (his/her full body or body's parts) in a three-dimensional space, (e.g., estimate the location of the object when it is lifted, and distinguish the location of the user's head when he/she is standing up or down)
- **Angular positioning (orientation)** can estimate the rotation and translation of an object or user (his/her full body or body's parts) in a three-dimensional space, (e.g., estimate the location of the object when it is tilted, and differentiate the location of the user's hand facing up or down)
- **Relative positioning (distance, proximity, neighboring)** can estimate the relative location between two or more entities in a three-dimensional space, (e.g., estimate the relative location between object—object, object—user, user's hand—another user's body, etc.)
- **Indoor positioning system (IPS)** can estimate the 3D position of an object or user limited to an indoor space, (e.g., estimate the location of the object inside a box, and estimate the location of a user inside a room)
- **Local positioning system (LPS)** can estimate the 3D position of an object or user limited to the range of a local network. It is suitable for outdoor environments, (e.g., estimate the location of the object near to a specific tree in the park, or the location of a person near to a sand tray in the backyard)

The 2D positioning system is often used for tabletop interaction since it allows determining the position of the object located in a plane surface. In the ToyUI mapping, we found different technologies that enable gaining access to such data. First, the AR-markers may be used to determine the object's 2D positioning. For example, IR tabletops that can recognize the fiducial markers attached to the bottom of the toy figurines. In other words, it is possible to use a single technology to gather both 2D positioning and NPID of the objects. In the industry, most items explored the conductive materials that allow interaction with the touch-screen of smartphones and tablets. The French company *Volumique* developed a technology license for their conductive touchpoints that enable gathering both 2D positioning and NIPD using unique conductive patterns. This license currently supports toy figurines of the *Hasbro's Spellshot*, the *Marbotics*, and *Oniri Islands*.

Computing technologies mixed different positioning features to create specific sensors and devices. For example, 3D positioning, orientation, and relative positioning are standard features of the commercial game controllers like the *Nintendo's Wiimote* and *Joy-Con*, and the *Sony's PlayStation Move Motion*. Such devices use their internal sensors to collect their 3D positioning and orientation and use IR sensors to determine their relative position to the TV (the place where the IR emitter is located). These game controllers may use a combination of embedded sensors (e.g., the accelerometer, gyroscope, magnetometer, and barometer), which can be directly embedded in the ToyUI items. The *Hasbro's Furby Connect*, for example, uses similar sensors to distinguish when the toy's body is located upside down, and can provide audio feedback to the user by "complaining" about it.

The type of sensor will determine the accuracy of estimating the object or user location, and for ToyUI it can range from sub-millimeters to meters. For example, the *LeapMotion* sensor estimates a more accurate hand's location than a user holding a game controller, and this sensor is used in ToyUI for tracking the manipulation of both physical and real objects. The desired accuracy should determine which sensor is suitable for the ToyUI design. When using both IPS and LPS it is possible to acquire a few meters of tracking field, and its accuracy will depend on the number of access points, beacons, or other wireless emitters. For example, an IPS using low latency tracking sensors like the *OptiTrack* products may conquer a more accurate detection range of both objects and users, and it can also allow a precise motion tracking.

Different from GPS, both positioning systems (IPS and LPS) are entirely designed. This makes it possible to perform user tracking without collecting PD. The

ROX commercial platform is an example of a non-personal LPS for ToyUI. It consists of a local network of connected devices that can be placed freely in the environment, such as on the floor, in a tree or hidden under a chair. Accessories allow wearing these devices attached to a belt turning the *ROX* device into a user-tracking device. The users can play a zombie-tag game, in which one player is randomly assigned the "zombie" state ID. Then by detecting the relative position between two user devices, in a range of centimeters, another player's state ID may be "infected" by the "zombie" ID, turning them into a collective ID of "zombies". As demonstrated in the last example, by combing different positioning systems and NPID it is possible to design innovative ToyUI solutions.

4.3. Motion tracking

Once we establish the object and user tracking strategies used to collect both NPID and positioning system information, it is possible to estimate motion-tracking data. Moving physical bodies generate a variety of data, including speed and acceleration. External forces such as momentum, gravity, and atmospheric pressure influence these bodies. Under ToyUI, real-time motion tracking is used as a resource to regulate physical play activities and to determine user performance. Next, we present common motion tracking data that should be obtained for ToyUI design:

- **Relative motion (trajectory)** can estimate the real-time relative positioning of the object or user (his/her full body or body's parts) from the starting location to the final one over a given time, (e.g., determine if the object or the user's finger is sliding on the table, and distinguish if the user has started to walk away or run)
- **Circular motion (angular frequency)** can estimate the real-time radial position of the object or user (his/her full body or body parts) from the starting angular position to the final location in a given time, (e.g., it determines if the object is rolling on the floor, and estimates if the user's head is rotating from left to right)
- **Oscillation (vibration)** can estimate the real-time 3D position of the object or user (his/her full body or body parts) by referencing a central axis and a movement frequency in a given time, (e.g., it estimates if the object is shaking, and distinguishes when a user is jumping)
- **Momentum (pressure)** can estimate the external forces exerted in the object or user (his/her full body or body's parts) over a given period, (e.g., it determines if the object has been thrown, landed or collided with another object, and distinguishes when a user is touching a soft object (atmospheric pressure)
- **3D Kinematics** can measure the kinematic chain movements of the object or user (its full body or body parts) in a three-dimensional space, (e.g., it estimates the chain movement of a robotic arm, and evaluates the user's hand gestures)

Motion tracking combines NPID and positioning system data. This allows tracking physical play activities, which are essential for ToyUI design. The Fuzzy Flyers brand introduced two soft-toys (a bird and a dog) that are shaped like balls, and that can estimate relative motion using their internal sensors. Children can play games like the "broken egg" using the bird toy, and in such a game, the ToyUI can measure momentum data. It can distinguish if the bird has arrived to a user's hands or on the floor, and it can distinguish a soft landing from sudden movements and consequently determine if the "egg" state ID has changed to "broken" or not.

Moreover, the Fuzzy Flyers' dog toy offers a dancing game, in which the sensors can estimate if the toy is shaking, and its frequency is regulated by following audio instructions provided by the toy. The instructions may request the user to shake the toy up and down, with varying motion intensity. Motion tracking allows measuring the player's performance by comparing real-time movements with the expected data collection.

In the next section, we will describe NPD patterns to assist ToyUI makers in efficiently combining identification, positioning system, and motion tracking data with different play rules.

5. Data patterns for NPD planning

ToyUI design mixes play features from traditional toys and games. Such combination may result in multiple social and physical play modalities (e.g., social competition, collaboration, or parallel play, and physical manipulation or full body interaction). A core play feature that defines both interactivity and connectivity requirements for ToyUI design is the play rules. The play rules can be classified into two general types: open-ended rules and closed rules. What differentiates them is how these modalities regulate the play activities to determine its state machine.

Diversely, the open-ended rules introduce open or negotiable rules, which are determined by the users and by the object's inputs and outputs (I/O) [5]. For example, the Multimodal Mixers [11] are allowed to update their stated ID by neighboring with each other, such as transferring their color data, and by reacting to motion tracking, like blinking their internal LEDs

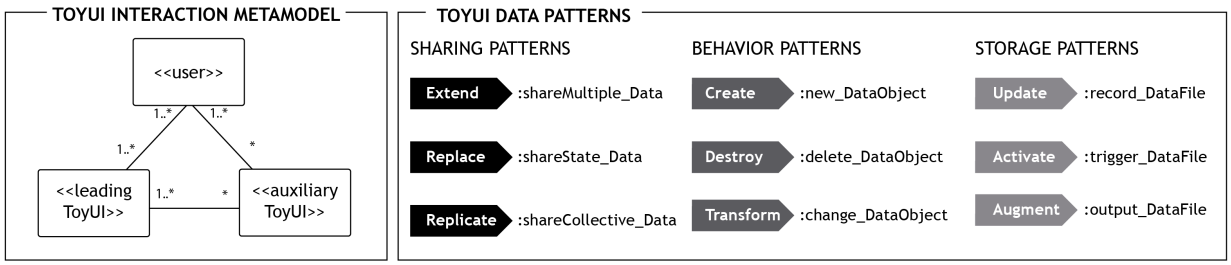

Figure 1. ToyUI interaction metamodeling and nine data patterns for NPD collection

when rolling, or changing color when shaking. Based on these I/O interactions, the children may create their play rules and games, such as assigning to the player's specific colors to regulate their play turns, or defining the winning and losing conditions. The degrees of freedom may be determined by the interaction rules that the ToyUI supports.

In contrast, the closed rules establish pre-defined rules to determine the game state machine, which enable the ToyUI designer to create levels and degrees of challenges that are similar to those used by digital games. This play modality may be associated with the toy-to-screen interaction, such as by combining physical inputs (the real toy) and digital outputs (the virtual world); or the opposite way, where the digital input may control the physical output (e.g., wireless controlled toys connected to their mobile applications).

A single ToyUI may introduce both play modalities. For example, the *Hasbro's Furby Connect* allows a user to engage in open-ended play by directly interacting with the robot-toy and its internal sensors, and in a closed rules setup by connecting the toy to the mobile application. In such an app, the toy can both send and receive data, update its state ID, and control the virtual object's behaviors.

We analyzed the data collection of each ToyUI item by observing the way the following operations were conducted: data sharing, behavior, and storage. We also examined how data usage was related to the play rules (open-ended and closed rules). This process resulted in the definition and identification of nine patterns for data collection planning. Figure 1 shows an interaction metamodeling for ToyUI design that is composed of the three entities: the *«user»*, the *«leadingToyUI»*, and the *«auxiliaryToyUI»* (e.g., toys, devices or peripherals). Along with the metamodeling, we introduce nine notations for each data pattern for use into UML-like diagrams in the data collection planning. Note that the nine patterns fall into three groups. The sharing group establishes an overall sharing mode that controls how data is transferred from one ToyUI component to another. The behavior group determines patterns that represent individual behavior that is regulated by the play rules. Finally, the storage group contains patterns reflecting the way data is managed by the ToyUI. Next, we will list each data pattern, its meaning, and provide an example of an application that used either open-ended or closed play rules:

- **Replicate** defines a collective data sharing between two entities (e.g., the user's single ID "zombie player" is replicated by another user's ID creating a collective ID "horde of zombies").
- **Extend** states a multiple data sharing between two entities (e.g., the object's "rope" ID is extended by a virtual object's "kite" ID creating a multiple ID to share momentum data).
- **Replace** outlines a state data sharing between two entities (e.g., the object's single ID "bird," is replaced by virtual object's state ID "egg" to share the object's relative motion data).
- **Create** describes the behavior of adding a new data object as result of data collection (e.g., the real object's relative motion creates the object's single ID "shooting" by lighting a red LED).
- **Destroy** expresses the behavior of deleting an existing data object as result of data collection (e.g., the object's momentum data destroys the virtual object's single ID "glass" by hitting on the projected wall).
- **Transform** defines the behavior of changing an existing data object as result of data collection (e.g., the user's 3D kinematics transforms the virtual object's prefab "fire ball" to another prefab "water ball").
- **Update** describes temporary storage of a data file to record a long-lasting action (e.g., the user's state ID "player," is updated to "team red" after attaching the card's single ID "red" to the belt).
- **Activate** states temporary storage of a data file to trigger a unilateral action (e.g., the object's radial motion activates the virtual object's stated ID "magic spell" by lighting the blue LED).
- **Augment** outlines temporary storage of a data file to output a specific action (e.g., the user's state ID "full health" is augmented by the virtual object's single ID "health" by lighting the green LEDs).

In the next section, we will show how to use these nine patterns and notations as part of two design solutions. Furthermore, we will propose a Printed

Circuit Board (PCB) architecture for ToyUI rapid prototyping that will support the NPD collection planning.

6. Rapid prototyping for NPD collection

A PCB architecture to support efficient NPD collection should meet the following requirements. Firstly, it must allow NPID collection related to the users, the leading ToyUI, and auxiliary ToyUI. Secondly, it should support both object and user tracking by acquiring 2D and 3D positioning, orientation, and motion tracking information. Thirdly, it must support LPS design by using both long and short-range wireless communication protocols. Finally, it should support open-ended and closed rules-based solutions by assuring both toy-to-toy and toy-to-device interaction.

Figure 2 illustrates the adopted PCB architecture. Its central unit (CU) is a microcontroller that has 14 digital I/O pins, a 16 MHz crystal oscillator, a serial to parallel interface (SPI), RS232 and I2C interfaces. The Wi-Fi module uses the RS-232 interface for communication with the CU; it has a low power 32 bit MCU, integrates a TCP/IP protocol stack, SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, and 4 GPIO. Additionally, it supports a wireless 802.11 b/g/n ISM Band interface and has a +20dBm output power when using the 802.11b wireless standard. The wireless proximity interface NFC/RFID module uses the SPI for communication with the CU. It is a highly integrated transmission module for contactless communication operating at 13.56 MHZ, and it supports four different operating modes. The reader/writer mode offers ISO 14443A / MIFARE®. The motion sensor module has an I2C Interface, a triple axis gyroscope with selectable scale (from ±250 to ±2000 dps), triple axis accelerometer with selectable scale (from ±2g to ±16g) and Digital Motion Processing™. We connected three RGB LEDs to three GPIOs to allow visual feedback. A buzzer links to the GPIO in order to provide audio feedback.

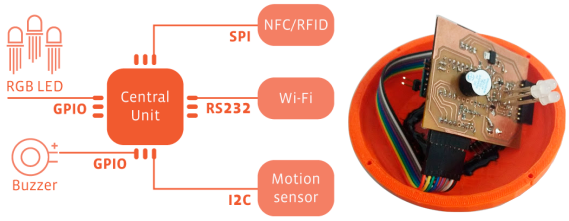

Figure 2. PCB architecture and solution.

Next, we will demonstrate two design solutions using our PCB architecture and show how we planned the adequate data patterns to design and implemented two ToyUI solutions. The first explored solution is based on closed rules, uses motion tracking, and the Wi-Fi interface to support toy-to-device interaction. The second solution combines the open-ended rules with NPID while using the proximity NFC/RFID interface.

6.1. Dragon ToyUI

The closed rules based solution focusses on supporting toy-to-device interaction. We embedded our PCB in a 3D-printed PLA (Poly-lactic Acid) rectangular box and attached to the back of a plush-toy using an elastic string (see Figure 3). With regard to our implementation, we had to modify the Wi-Fi module by adding an internal firmware to create a wireless access point that allowed us to send the motion sensor's data to the mobile application. The mobile application was developed using the game engine *Unity 3D*, and it consisted of a list of 2D animations associated to each state ID. These were an "idle loop" animation and two animations for each play action (i.e., "roar" and "fire" animations). All *«user»* inputs were designed to be performed in the *«leadingToyUI»*, and all visual and audio feedback was designed for output in the *«auxiliaryToyUI»*.

Figure 3. Dragon ToyUI prototype setup.

Figure 4 shows how we planned the NPD collection using the ToyUI interaction metamodeling by selecting the adequate data patterns from the list. We used a planning diagram that incorporated UML-like notations. It consists of a mixture of features from the UML's class, sequence, and activity diagrams. In the diagram, we describe the sequence of data collection relating to the three entities: the player (*«user»*), the dragon-toy (*«leadingToyUI»*), and the smartphone (*«auxiliaryToyUI»*). The diagram portrays the data patterns notations used to describe the data sharing, behavior, and storage.

Firstly, we planned the data sharing since it would define how collected data collection would then be transferred from the *«leadingToyUI»* to the *«auxiliaryToyUI»*. We selected the *«replicate»* pattern so that the dragon-toy ID and the virtual dragon-prefab ID would constitute of a collective ID. Secondly, we planned the individual behaviors by relating them with the set of closed rules. Hence, the *«user»* would

«*create*» new circular motion data in either the X or Z axis. As a result, of data collection, the existing dragon-prefab would «*transform*» its shape. Finally, we planned how the ToyUI manages the data storage. It explores the three available data patterns: the «*update*» was used to record the "idle" state ID of the dragon-prefab, the «*activate*» was used to trigger temporary state changes of the state ID (e.g., "fire" and "roar" IDs), and the «*augment*» was used to output feedback to the «*user*» by displaying individual animations.

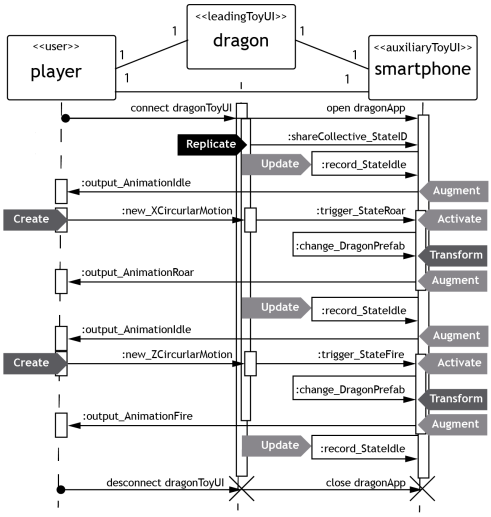

Figure 4. Dragon ToyUI NPD planning.

6.2. Smartminton

The open-ended rules solution focusses on supporting screenless toy-to-toy interaction. We designed a «*leadingToyUI*» that allowed physical exertion and that could regulate the play rules. The *Smartminton* is a smart shuttlecock toy-prototype (see Figure 5). Its body was 3D-printed using the flexible material *NinjaFlex*, which is made of thermoplastic polyurethane. We used it to reduce the internal physical impact during collisions. In addition, we covered the full body with a yellow fabric bag filled with acrylic stuffing. We designed the body in two separated faces that were attached by four tiny screws so that we could embed our PCB and other components. The feathers were made of layers of white organza fabric that were manually waterproofed using a white glue solution. We choose this thin fabric to allow the RGB LEDs to reflect the feedback color on the feathers. To allow the user's tracking, we designed four colorful rubber gloves augmented by individual NFC tags.

Figure 5 also illustrates the open-ended rules designed for this ToyUI. It consisted of a primary rule: the current LED color would define the player's turn by matching it with the color of its glove. Moreover, it contains three forbidden actions: do not "drop" or "hold" the toy, and the «*user*» cannot "unmatch" its color turn. Only the "unmatch" rule was regulated by the «*leadingToyUI*» since open-ended play experiences are co-located social play experiences, and some rules must be regulated or negotiated by the players themselves, as a means to stimulate social interaction.

Figure 5. Smartminton setup and play rules.

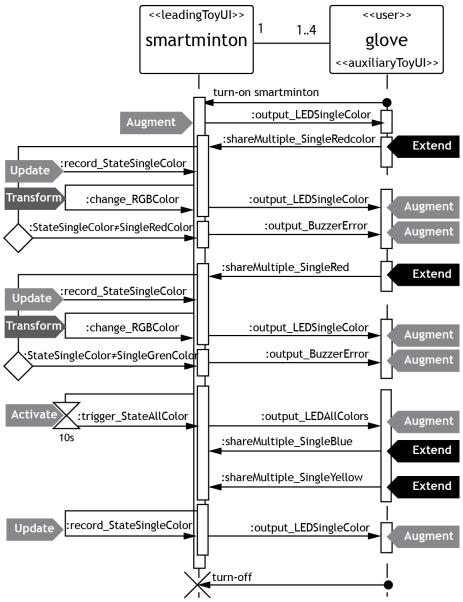

Figure 6. Smartminton NPD planning.

Figure 6 shows the NPD collection planning of the *Smartminton* (*«leadingToyUI»*). In this scenario, each player wears a glove. Thus, we planned the data collection by combing the *«user»* and the *«auxiliaryToyUI»* into a single entity of the diagram. We planned the *Smartminton* data collection to use only NPID. Thus, the *«leadingToyUI»* used the NFC module to identify each glove. We selected the *«extend»* pattern to establish the data sharing. Each glove would *«extend»* its single ID (i.e., either "red," "yellow," "blue," or "green" IDs) to request multiple data sharing with the *«leadingToyUI».* As a result, the *«leadingToyUI»* would *«update»* its state ID, and then, would verify if either of the multiple IDs was a match or not. After every extension attempt, the existing RGB color would *«transform»* its data, and the *«leadingToyUI»* would *«augment»* the next turn by

outputting a single LED color. If the colors were unmatched, it would additionally *«augment»* an error message sound, emitted by the buzzer. Moreover, we defined an additional state ID, the "all colors" ID which was *«activated»* every 10 seconds. Such a state ID allowed all players to freely *«extend»* their multiple IDs.

7. Conclusion and future work

ToyUI solutions may collect both PD and NPD from their users. NPD collection for ToyUI can be achieved efficiently, as a means to create innovative and meaningful play experiences that respect user privacy. We proposed an NPD collection planning approach that explores nine data patterns for ToyUI design under both open-ended and closed play rules. We demonstrated the application of these nine patterns by introducing a planning diagram using UML-like notations. We used it to plan two ToyUI solutions that embed the same PCB architecture for rapid prototyping. The PCB requirements were elicited based on the data collection of 297 items from both the literature and industry mappings. The proposed PCB architecture allowed the collection of NPID, different positioning systems (e.g., 3D-positioning, orientation, and LPS), and motion tracking. It further supported interoperability by combining Wi-Fi and RFID/NFC protocols.

Future versions of the PCB may include improvements in the motion sensor by adding an atmospheric pressure component to amplify the momentum's data tracking. Moreover, we could replace the buzzer with an 8-bit speaker, and add a vibrating motor to provide haptic feedback to the users. Future works could cover both user testing of the solutions, and performance tests to assess disruptions due to the presence of noise. Other topics include the construction of datasets benchmark for NPD and play analysis, and the establishment of the MQTT or other universal protocols for the collection, organization, and storage of data. Ultimately, this work can be used to guide future ToyUI projects for both toy makers and researchers. We expect that our strategies may encourage them to use efficient NPD collection and planning, by assisting them in designing solutions that are safe (and pleasant) to use and play with.

8. Acknowledgments

This research was partially supported by CNPq in the scope of IMPReSS project (490075/2013-4) and Ph.D. scholarship supported by FACEPE.

9. References

[1] L. Rafferty, B. Kroese, and P. C. Hung. "Toy computing background". Mobile Services for Toy Computing. Springer, Cham, 2015. 9-38.

[2] L. Rafferty, P. C. Hung, M. Fantinato, S. M. Peres, F. Iqbal, S. Y. Kuo, and S. C. Huang. "Towards a Privacy Rule Conceptual Model for Smart Toys". Computing in Smart Toys. Springer, Cham, 2017. pp. 85-102.

[3] D. Locke, "Mq telemetry transport (mqtt) v3. 1 protocol specification". IBM developerWorks Technical Library, available at http://www. ibm. com/developerworks/ webservices/library/wsmqtt/index. html, 2010.

[4] M. Schmitz, J. Steimle, J. Huber, N. Dezfuli, and M. Mühlhäuser. "Flexibles: Deformation-Aware 3D-Printed Tangibles for Capacitive Touchscreens". Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. ACM, 2017. pp. 1001-1014.

[5] I. Soute, S. Lagerström, and P. Markopoulos. "Rapid Prototyping of Outdoor Games for Children in an Iterative Design Process". Proceedings of the 12th International Conference on Interaction Design and Children. ACM, 2013. pp. 74-83.

[6] A. P. de Albuquerque. "IoT4Fun: A Relational Model for Hybrid Gameplay Interaction". Master Thesis. Universidade Federal de Pernambuco. Brazil, 2017.

[7] B. Kitchenhan, S. Charters. "Guidelines for Performing Systematic literature reviews in Software Engineering." Technical report, Ver. 2.3 EBSE Technical Report. EBSE. sn, 2007.

[8] A. Melonio and M. Rizvi. "The Design of Turntalk for the Scaffolding of Balanced Conversations in Groups of Children". International Symposium on Emerging Technologies for Education. Springer, Cham, 2016. pp. 278-287.

[9] S. Tobias and M. M. Jensen. "ChillFish: a Respiration Game for Children with ADHD". Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction. ACM, 2016. pp. 271-278.

[10] D. Avrahami, J. O. Wobbrock, and S. Izadi. "Portico: Tangible Interaction on and Around a Tablet." Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 2011. pp. 347-356.

[11] T. Bekker, J. Sturm, and B. Eggen. "Designing Playful Interactions for Social Interaction and Physical Play". Personal and Ubiquitous Computing 14.5. 2010. pp. 385-396