University of Birmingham
Research at Birmingham

A Tangible End-User Development tool supporting the repurposing of Pervasive Displays

Turchi, Tommaso; Malizia, Alessio; Dix, Alan

DOI:
10.1016/j.jvlc.2016.11.002

License:
Creative Commons: Attribution-NonCommercial-NoDerivs (CC BY-NC-ND)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Turchi, T, Malizia, A & Dix, A 2016, 'A Tangible End-User Development tool supporting the repurposing of Pervasive Displays', *Journal of Visual Languages & Computing*. https://doi.org/10.1016/j.jvlc.2016.11.002

Link to publication on Research at Birmingham portal

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.
•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
•User may use extracts from the document in line with the concept of ‘fair dealing’ under the Copyright, Designs and Patents Act 1988 (?)
•Users may not further distribute the material nor use it for the purposes of commercial gain.

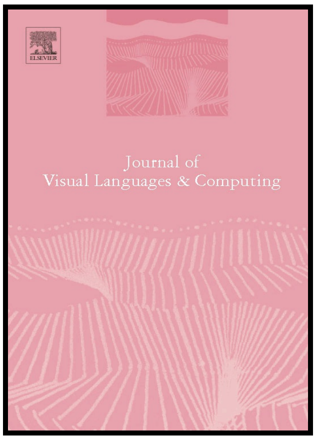
Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 24. Apr. 2024



www.elsevier.com/locate/jvlc

TAPAS: A Tangible End-User Development tool supporting the repurposing of Pervasive Displays

Tommaso Turchi, Alessio Malizia, Alan Dix

PII: S1045-926X(16)30219-1
DOI: http://dx.doi.org/10.1016/j.jvlc.2016.11.002
Reference: YJVLC767

To appear in: *Journal of Visual Language and Computing*

Received date: 1 December 2015
Revised date: 24 October 2016
Accepted date: 1 November 2016

Cite this article as: Tommaso Turchi, Alessio Malizia and Alan Dix, TAPAS: A Tangible End-User Development tool supporting the repurposing of Pervasiv Displays, *Journal of Visual Language and Computing* http://dx.doi.org/10.1016/j.jvlc.2016.11.002

TAPAS: A Tangible End-User Development tool supporting the repurposing of Pervasive Displays[☆]

Tommaso Turchi[a,*], Alessio Malizia[a], Alan Dix[b,c]

[a] *Human Centred Design Institute, Brunel University London, London, UK*
[b] *HCI Centre, School of Computer Science, University of Birmingham, Birmingham, UK*
[c] *Talis, Birmingham, UK*

Abstract

These days we are witnessing a spread of many new digital systems in public spaces featuring easy to use and engaging interaction modalities, such as multi-touch, gestures, tangible, and voice. This new user-centered paradigm — known as the Natural User Interface (NUI) — aims to provide a more natural and rich experience to end users; this supports its adoption in many ubiquitous domains, as it naturally holds for Pervasive Displays: these systems are composed of variously-sized displays and support many-to-many interactions with the same public screens at the same time. Due to their public and moderated nature, users need an easy way of adapting them to heterogeneous usage contexts in order to support their long-term adoption. In this paper, we propose an End-User Development approach to this problem introducing TAPAS, a system that combines a tangible interaction with

---

[☆] This is an extended and revised version of a paper that was presented at the 2015 Symposium on Visual Languages and Human-Centric Computing [1]. This paper significantly expands over TAPAS' design rationale, presentation, and resulting discussion.

[*] Corresponding author

Email addresses: `tommaso.turchi@brunel.ac.uk` (Tommaso Turchi), `alessio.malizia@brunel.ac.uk` (Alessio Malizia), `alan@hcibook.com` (Alan Dix)

a puzzle metaphor, allowing users to create workflows on a Pervasive Display to satisfy their needs; its design and visual syntax stem from a study we carried out with designers, whose findings are also part of this work. We then carried out a preliminary evaluation of our system with second year university students and interaction designers, gathering useful feedback to improve TAPAS and employ it in many other domains.

Keywords: End-User Development, Tangible User Interfaces, Natural User Interfaces, Pervasive Displays, Tangible Programming

1. Introduction

In the past few years our lives have been flooded by a multitude of new ubiquitous computing systems, including multi-touch-enabled smartphones, voice-controlled virtual personal assistants, gesture-recognition devices, and so on. These systems all feature a revolutionary emerging interaction paradigm evolved over the past two decades and founded on the most basic and innate human interaction capabilities, such as touch, vision and speech, known as Natural User Interfaces (NUIs). Unlike the more traditional interfaces based on artificial control mechanisms such as the mouse and keyboard, a NUI relies only on a user being able to carry out simple and arguably easily discoverable motions to control the on-screen application and manipulate its content.

Tangible User Interfaces (TUIs) represent one of the first successful attempts at developing a NUI, inspired by the physical world, thus allowing users to interact with the digital system in the same way as they would interact with a physical object, providing data and computational power with a physical shape [2]. By taking advantage of our innate dexterity for object manipulation, TUIs have proven to be very effective in making highly abstract activities such as programming more direct and accessible. In addition, there are interesting preliminary results linking the usage of tangible tools with increased ability to model abstract concepts [3, 4]; these findings suggest that physical manipulation acts as a scaffold between the real and digital, enhancing Computational Thinking skills [5].

Nevertheless, it is not just the input modality that makes an interface natural: it has to leverage each user's capabilities and meet her needs while fitting the current task and context demands. The design of innovative digital systems like Pervasive Displays has indeed followed many principles [6] with the aim of making interactions as *natural* as possible, in order to support their appropriation and widespread use; these systems are composed of various-sized displays supporting a many-to-many interaction modality and allowing "many people to interact with the same public screens simultaneously" [7].

In recent years, thanks to the newly available technologies and the intuitive interaction capabilities of Pervasive Displays, they have spread around public areas such as museums, tourist information centers, universities, shopping malls and various other urban locations [8, 9]. A new trend has recently emerged within Pervasive Displays research, namely to design large and long-term deployments outside traditional controlled laboratory settings with no researchers' supervision, in other words *in-the-wild*. These studies evaluate artifacts in their habitual use context within people's lives; this means observing and recording what people do with them and how their usage changes over time [10].

Yet, long term deployments of Pervasive Displays present two main drawbacks [11]: (1) their setup and daily operational activities are expensive, and (2) users and site managers tend to lose interest in their usage and maintenance over time. Even if at first it is easy to advertise the provided benefits through articles and papers presenting similar success stories, problems start to surface when the initial buzz and enthusiasm (novelty) wears off and managers have to carry out the daily maintenance tasks. In addition, keeping these systems interesting over time by constantly providing them with fresh content has proven to be particularly challenging [12].

To solve these problems, Hosio et al. [11] suggest that allowing a degree of appropriation when designing Pervasive Displays might enable all the stakeholders to understand how such systems could relate to the ordinary activities they often take for granted, leading to a more sustained and prolonged use. Moreover, their public and moderated nature does not allow the provision of a broad set of general purpose and unfixed features, because users' interests and needs are commonly heterogeneous and continuously evolving. Thus, Pervasive Displays need to be adapted to all the different users' needs and repurposed as those needs shift over time to promote a more serendipitous and prolonged usage.

We then argue that End-User Development (EUD) could be effective in enabling users to adapt and repurpose Pervasive Displays without any intervention of site managers. In addition, in order to provide a coherent and immersive user experience, users need to be able to carry out this activity in the most natural way possible, ideally in the same way they already interact with existing Pervasive Displays; for this reason, we are advocating the use of a TUI to carry out their repurposing.

The contributions of this paper are twofold. First, we present an application for Pervasive Displays, combining a TUI — employing the user's smartphone as the physical probe — with a visual interface projected onto a tabletop display; our prototype — called TAngible Programmable Augmented Surface (TAPAS) [13, 1] — aims at providing users with an easy and simple way of composing simple task-oriented applications (e.g., downloading a PDF from the user's Dropbox account and displaying its preview on the main tabletop screen). Second, we highlight some of the main challenges faced by Tangible Programming on Pervasive Displays stemming from two preliminary studies we carried out with end users and designers.

In particular, we outline the process we went through in designing TAPAS, whose interaction paradigm stems from the results of a workshop we carried out with expert designers, which we used to collect insightful ideas and design challenges related to the introduction of an EUD metaphor to a tangible interactive tabletop. Our application is designed to foster collaboration and support appropriation of Pervasive Displays systems in many different contexts of use (e.g., within a company to support users creating and sharing data analyses); the first evaluation scenario we have selected is within an educational space to foster students' collaboration on different projects during their recurring group meetings. To validate the efficacy of the proposed interaction for guiding users in the composition of different applications, we carried out two preliminary formative evaluations within a collaborative work scenario, involving, respectively, second year university students working on a group project and interaction designers. Strictly speaking, since this study is not completely in-the-wild, the findings can only be leveraged to improve the design of TAPAS, thus further studies are needed to draw more definitive conclusions over the proposed interaction modality within this scenario.

2. Related Works

2.1. Tangible User Interfaces

Declining hardware costs have recently enabled many new technologies to be available to a wider audience, together with new and engaging interaction modalities, particularly using gestures or object movements; this revolutionary paradigm goes under the name of the Natural User Interface (NUI), and it allows people to act and communicate with digital systems in ways to which they are naturally predisposed.

The term ‘natural’ has been used in a rather loose fashion, meaning intuitive, easy to use or easy to learn; many studies argue that we can design a natural interaction either by mimicking aspects of the real world [14] or by drawing on our existing capabilities in the communicative or gesticulative areas [6].

One of the most successful and developed approaches falling into the first category has been introduced by Ishii et al. [2] and is known as Tangible User Interfaces (TUIs). The aim of TUIs is to give bits a directly accessible and manipulable interface by employing the real world, both as a medium and as a display for manipulation; indeed by connecting data with physical artifacts and surfaces we can make bits tangible.

Many studies in this research area investigate the supposed benefits offered by this interaction paradigm, ranging from intuitiveness [2], experiential learning through direct manipulation [15, 16], motor memory [17], accuracy [18], and collaboration [19]. Furthermore, the effects of employing a TUI to interact with a digital system are certainly dependent on the tasks and domain, as many comparative studies suggest [17, 18, 20]; for this reason, Kirk et al. [21] made the case for hybrid surfaces, employing physical elements together with digital ones.

Researchers are also debating how employing TUIs reflects on learning [4, 22, 23], with specific reference to highly abstract concepts: this stems from Piagetian theories supporting the development of thinking — particularly in young children — through manipulation of concrete physical objects. Other studies [5, 24] are even linking this effect to the development of Computational Thinking skills [25], namely a new kind of analytical thinking integral to solving complex problems using core computer scientists' tools, such as abstraction and decomposition.

Due to the ubiquitous nature of our scenario and the aforementioned traits of TUIs, we felt that designing our system around a tangible interaction would contribute to fostering its usage in a more sustained and prolonged way.

2.2. End-User Development

The End-User Development (EUD) research community always strives to make programming tasks easier for end users (any computer user), thus allowing them to adapt software systems to their particular needs at hand. Visual Programming (VP) is one of the most well-studied techniques, aiming at lowering barriers and often used to first introduce to coding: it reduces the traditional syntactic burden of a programming language (often domain-specific — i.e., tailored to a given application domain) by encapsulating it with a visual representation of its instructions, using graphical tweaks to communicate the underlying semantic rules at a glance.

Programming components can, for example, be represented by different blocks, allowing users to combine them together to build a working program. Constraints over different data types can be enforced by using different shapes and allowing only matching inputs and outputs to be combined. Using blocks to represent program syntax trees is a recent trend in designing VP systems [26], as witnessed by the spread of Block Programming Environments like Scratch[1], Microsoft Touch Develop [27], App Inventor[2], and MicroApp [28], to name but a few.

Yet another way of aiding users in their programming task is by employing tangible objects. The existing literature can be clustered in two main categories according to the paradigm employed: Programming by Demonstration (PbD) or Programming by Instruction (PbI). PbD, also known as Programming by Example, enables users to teach new behaviors to the system by demonstrating actions on concrete examples [29]. PbI, known as Tangible (sometimes Physical) Programming within the TUI domain, takes a traditional approach to programming, that is requiring users to learn and employ a syntactic construct (e.g., text instructions, natural or visual languages) to impart instructions to the system.

Topobo [16] — proposed by Parkes et al. — falls under the first category and comprises a set of components that one can assemble and animate with

---

[1] `https://scratch.mit.edu`

[2] `http://appinventor.mit.edu`

different manipulations; one can then observe the system repeatedly play those motions back. PbD proved to be an effective and intuitive way of teaching different movements to a system directly on actuated physical objects, therefore it has been specifically named Robot Programming by Demonstration [30]. The system devised by Lee et al. [31] uses a different approach: this PbD system allows users to record macros composed by physical and digital actions performed on several objects, such as opening a drawer, turning on the TV, and so on; the system records the actions' sequence and plays them back in the same order once the first action is performed.

These systems offer an unparalleled experience in terms of ease of use, but — due to the paradigm they employ — present a quite substantial limitation: users can interact only with the outputs, therefore the instructed behaviors are necessarily composed solely of operations that are directly available, resulting in the inability to represent more complex behaviors; this is the reason why the main problem of PbD systems is the generalizability — i.e. finding the general semantics — of instructed behaviors [29].

Moving to PbI-based systems, Mugellini et al. [32] proposed the concept of tangible shortcuts: they improved information access and retrieval using physical objects, enabling users to develop new shortcuts through a Visual Language based on a puzzle metaphor. In 2012 Wang et al. introduced E-Block [33], a tangible programming tool for young children, enabling them to instruct a robot's movements by assembling different blocks, each assigned to a specific function. Robo-Blocks is a similar system presented in the same year by Sipitakiat and Nusen [34], which added the ability for users to debug their applications using a display placed on top of each block.

However, the majority of Tangible Programming systems keep the digital and physical perspectives completely detached from each other: tangible objects are assembled together based on their own physical features to create digital constraints on the final program, while their digital representation is separated to be shown and used (i.e., executed) only later. Our platform joins these two perspectives, using physical and digital constraints together to make the experience as smooth as possible and give more flexibility to the whole system.

2.3. Pervasive Displays

In the last two decades Pervasive Displays — namely ecosystems of various-sized displays supporting simultaneous interactions with the same public screens [7] — have become common within public areas. As well as attracting the general public's interest, this has led to the flourishing of an active research community[3]. Most of the studies within this area are carried out in-the-wild [35], in other words outside of controlled settings, in places where such systems are commonly to be found.

One of the main problems within this research area is personalization [36]: due to their open nature and ubiquitousness, designing a Pervasive Display to fit every user's needs — users who often take on different roles [37] — is quite challenging, thus many efforts have been made to find ways of adapting their features to different contexts of use with different degrees of automation, including allowing users to design a system's components themselves at use time.

---

[3] `http://pervasivedisplays.org`

Cremonesi et al. [38] employs both a touch and touchless interaction paradigm on a large public display in order to offer a personalized experience to users, based on their profile. The system is capable of recognizing single users, couples or even groups and can be paired with users' smartphones in order to achieve a higher degree of personalization. Exploiting users' smartphones to personalize the system is a widely used technique in the area [39]: Tacita [40] is yet another Pervasive Display system that draws personalized content from users' smartphones and displays it on the big screen.

Using the main screen as a centralized hub to display and share resources is also the main idea behind Dynamo [41], which allows users to exchange digital resources with each other using PCs, USB sticks and PDAs, as well as viewing and annotating them collaboratively.

The high level synergy between Pervasive Displays and Personal Devices such as smartphones has been well modeled in the design framework introduced by Dix and Sas [42]: the many roles to be taken by a personal device are described in relation to a public situated display, e.g., it may be a selection or pointing device, or a personal identification trigger. To the best of our knowledge, though, existing systems exploit smartphones merely as containers of users' information to be crawled to personalize the big screen or as simple selection devices [43], rather than considering them as fully-fledged tangible objects, whose shapes — either physical or digital — and movements can afford their available interactions; the sole exception is mentioned by Dalsgaard and Halkov [44], who are thinking of introducing smartphones to their tabletop system based on tangible interaction, in order to afford individual and more complex interactions on such devices together with collaborative interactions on the main tabletop.

Our system's design exploits smartphones both as personalization triggers and as interaction control mechanisms, in order to leverage the benefits brought in by having a tangible interaction and an adaptable Pervasive Display.

3. TAngible Programmable Augmented Surface

The aim of the proposed prototype, called TAngible Programmable Augmented Surface (TAPAS), is to allow users to adapt the features offered by a public interactive display through Tangible Programming. This combines End-User Development (EUD) with a Tangible User Interface (TUI) instead of a classic GUI-based Visual Language, exploits Meta-Design principles to foster appropriation [45], and allows users to become designers themselves, by empowering them to adapt the system to their own specific needs.

We began TAPAS' development by carrying out a workshop with experts to explore the challenges and opportunities of our design space; we collected ideas and suggestions that have then been used to drive the design.

3.1. Design

The design of TAPAS aims to provide users with a tool to assist them in solving simple tasks in various collaborative work scenarios, as we stated in the introduction. It is our attempt at fostering long-term sustained appropriation of Pervasive Displays by enabling users to repurpose the system themselves through EUD.

TAPAS' programming environment uses a Block-based Programming approach [46] — widely used by systems like Scratch [47] and Blockly[4] — that has proven to have a low learning threshold for non-programmers.

TAPAS allows users to create, share, modify and reuse simple workflow applications, namely sequential processes combining different services in a data-flow fashion, where the output of a service becomes the input of the following one. Indeed, we noted that in public displays the majority of applications provided are in the form of services that ideally can be combined to satisfy specific users' needs. For example a public display may provide different services for tourists in which it might present a specific guide to a city with some information about events or points of interest. Currently, these services are normally not linked and users cannot combine them to build a new service that might better suit their needs.

Users impart instruction to TAPAS through a visual syntactic construct in a Programming by Instruction (PbI) fashion rather than by demonstrating their intentions to the system: indeed, making a workflow's inner architecture transparent to users will allow them to better understand its sequential logic and behavior, improving their skill in using the system.

Our system's blocks — represented either digitally or physically — correspond to workflow components (i.e. functions) that users can assemble together as in other Block-based Programming environments; each block receives specific formats of data as input and produces different ones as output based on its inner workings and its location within a workflow's logic.

---

[4] `https://developers.google.com/blockly`

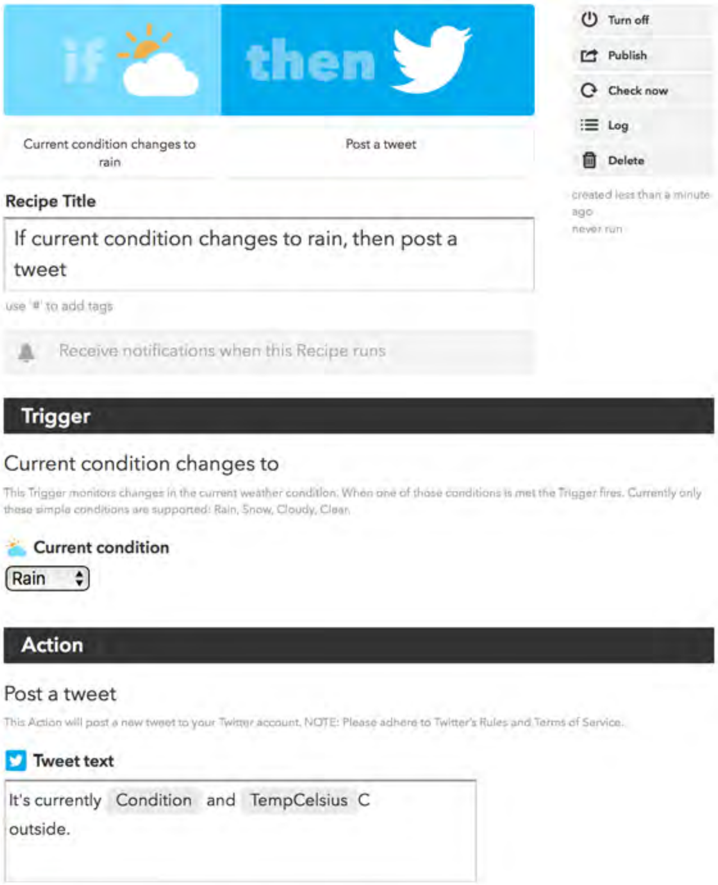
To devise a syntax that focuses on simplifying workflow development for users and effectively integrate a Block-based Programming approach with a TUI on a tabletop, we carried out a workshop with experts to better understand the design space. We gathered five experts with backgrounds in different design areas for a one-hour focus group in a university meeting room: three experienced interaction designers with some basic programming knowledge — one with a specific background on information visualization and one with quite substantial industry experience — and two product designers without any programming experience at all.

During the workshop's first phase — lasting 30 minutes — participants were instructed in the context of this research and the specific scenario we are focusing on. We showed them some examples of workflows from IFTTT (IF This Then That)[5], a widely popular Web mashup system [48]; it allows users to create simple event-based *if-then*-style workflows with different Web services and acts as a hub connecting their events' triggers with actions: one can describe simple rules by selecting the event that will trigger the workflow (e.g., when the current temperature rises a provided value or when the user edits a specific file on Dropbox) and an action that should be performed in any other — even the same — supported Web service (e.g., tweet about it or send the file via email), as shown in figure 1. We have used these examples to showcase different types of workflows, their inner logic and how the trigger selection provides the subsequent action with anchors dependent on the output's type: when the event concerns a location the action can

---

[5] `http://ifttt.com`

access its GPS coordinates, when it involves a text file the action will be able to use its content, and so on.



Current condition changes to rain

Post a tweet

created less than a minute ago
never run

Recipe Title

If current condition changes to rain, then post a tweet

use '#' to add tags

Receive notifications when this Recipe runs

Current condition changes to

This Trigger monitors changes in the current weather condition. When one of those conditions is met the Trigger fires. Currently only these simple conditions are supported: Rain, Snow, Cloudy, Clear.

Current condition

Rain

Post a tweet

This Action will post a new tweet to your Twitter account. NOTE: Please adhere to Twitter's Rules and Terms of Service.

Tweet text

It's currently Condition and TempCelsius C outside.

Figure 1: An example of a workflow created using IFTTT: when the condition in the user's location changes to rain (trigger) it will automatically post a tweet (action).

We then showed participants a video about an existing TUI system — the Tangible 3D Tabletop [44] — summarizing the benefits of this interaction paradigm; in particular, we highlighted the different metaphors involved in tangible systems, in relation to the physical and the digital domain [49].

After the introduction, participants started a 30-minute discussion about ideas and challenges for the design of TAPAS' syntax, focusing on a collaborative work scenario involving users with no previous programming experience.

3.1.1. Preliminary Findings

The features that suggested participants should be included in TAPAS were clustered based on their domain: they either concern TAPAS' tangible objects or its digital syntax. Here are the main findings from the workshop:

Tangible features. Participants stressed the fact that the system should react only upon user actions and provide useful feedback through a specific communication channel, in agreement with one of the main principles of Natural User Interfaces (NUIs) [50]. Many suggestions focused on the preferred channel to be used to provide feedback. These included equipping tangible objects with a touch-sensitive mechanism in order to activate the feedback only when users physically touch objects on the table, in order to highlight whether selected objects are compatible with each other (fulfilling the workflow constraints). Moreover, the communication channel of choice can be a physical one as well: a magnetic attraction between objects could indicate when two workflow's components are compatible with each other, while repulsion might represent the opposite. Another participant suggested employing haptic feedback built into the tangibles to communicate compatibility between different ones.

Digital features. Another set of suggestions were directed towards the digital representation of our platform's syntax. First, the blocks' digital representation could help users understand components' constraints by using, respectively, different and similar colors or shapes for incompatible and compatible components. Also, since a workflow's composition is usually performed one component at a time, i.e. by selecting a function that will follow the latest assembled one, our system might aid users on the next available components to be chosen by changing the color or the shape of the currently assembled workflow. Lastly, since TAPAS shows all available components at once, this gives the user an overall view of the system's capabilities. However, this also allows users to make mistakes. TAPAS is intended to be used by inexperienced users, so we need to assist users in finding the right way of assembling different components, when they cannot figure it out themselves; a useful suggestion in this regard is to provide some sort of "translation tool", which — once a user selects two blocks incompatible with each other — shows them at least one possible way of choosing other components in between to connect the two blocks, assisting users during the composition phase.

After collecting these suggestions from the workshop, we designed TAPAS trying to fulfill the majority of them; we present the details of its implementation in the following section.

3.2. Architecture

TAPAS comprises a horizontal tabletop display and an RGB camera capturing the movements of the users' smartphones on the main display's surface using fiducial markers [51] (i.e. images used as a point of reference when placed in the camera's field of view), as summarized in figure 2; it supports the Tangible User Interface Objects (TUIO) protocol [52], already adopted by many research communities within the TUI area as a general and versatile communication interface between tangible tabletop controller interfaces and underlying application layers, which has been designed specifically for interactive multi-touch tabletop surfaces.

When a user logs into our web application running on a smartphone using her credentials, this will display a fiducial uniquely assigned to that account. The system can then track the position of the fiducial across the tabletop surface, knowing to whom it belongs; hence, smartphones represent objects whose movements allow users to interact with the system, i.e. they form the physical and digital representation of information in our system, and are already equipped with all the sensors and feedback mechanisms needed to implement the designers' suggestions obtained from the workshop. We are exploiting smartphones to adapt the system to the different users' preferences because they hold much of the users' personal information — such as their Facebook and Dropbox login credentials. Moreover, this will protect users' privacy by sharing only the minimum set of information required to set up a service (users are in control of privacy settings) and the smartphone can be used to display a wide range of widgets that can be presented to end users depending on the specific service being accessed (e.g., a virtual keyboard to input text).

Finally, portable devices can also be used to store the outputs created by end users, having a multiple positive effect: users will be able to carry with them the outputs of the applications created on a public display for later use, and also the use of a mobile device can mitigate network failures by supplying personal data stored on the device itself.

3.3. Interaction Paradigm

In order to simplify workflow development for end users, we have used the metaphor of recipes: a recipe is a workflow performing a particular task and

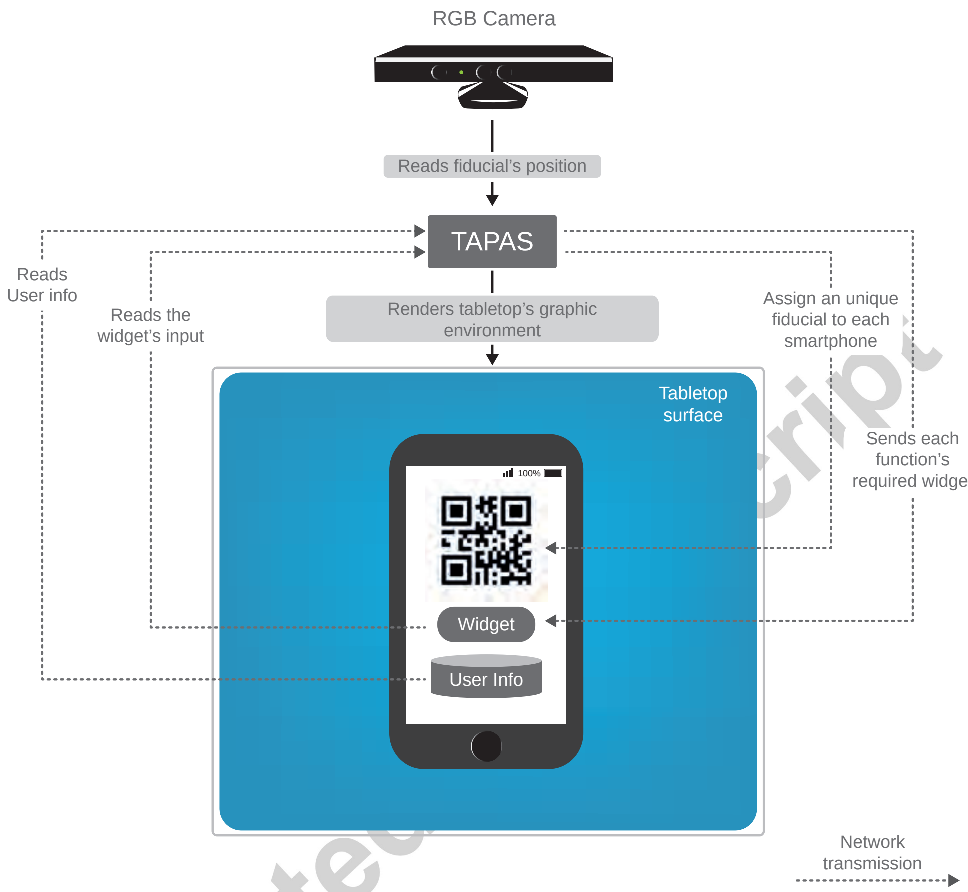


Figure 2: The architecture of TAPAS: using a fiducial marker — assigned by the application itself — and a RGB camera, TAPAS can track a smartphone's movements on a tabletop surface; through the smartphone, TAPAS is able to link each and every smartphone's movements to its users and display a corresponding dynamic widget.

is composed of different functions — or ingredients; moreover, a recipe can become a service itself, thus it can then be included in other recipes, fostering their reuse. In the future, users will be able to share their recipes or modify the ones they or others have created, just as they do with real recipes in their cookbooks. Thanks to the introduction of this recipe mechanism our prototype allows users to share services with others who might have the same needs. Furthermore, as would happen in real life, if someone does not have a specific ingredient for a recipe she would seldom change recipe but instead find a way of replacing an ingredient with one that is available, in agreement with the results of the design workshop, which suggested providing some sort of "translation tool" to help users finding missing components needed to join two blocks. Moreover, if, for example, a service is not available due to network failure, our recipe metaphor and the use of a smartphone still allows data stored locally on the device to be used in services included in the recipe.

We have used a puzzle metaphor to communicate basic control-flow sequentialization mechanisms since such a metaphor is quite familiar to end users and should ease the workflow editing [53]: each puzzle piece represents an available function (or ingredient, carrying on with the recipe metaphor) which could require some inputs and produce some outputs, as depicted in figure 3; type constraints on different inputs and outputs are afforded using different shapes. The smartphone itself is associated with the main puzzle piece, a circle halo with a single hollow to accommodate the next piece, which will move alongside the smartphone on the main display's surface; moving the main piece towards another one will add the latter's related function to the workflow — if the two shapes are matching, that is to say the latest output is compatible with the required input. This helps end users to understand the data-flow approach as well as type constraints. If a single piece requires some additional inputs from the user, such as selecting one option from several, or typing in some text, a dynamic widget will appear on the lower half of the smartphone screen, allowing the user to do so.

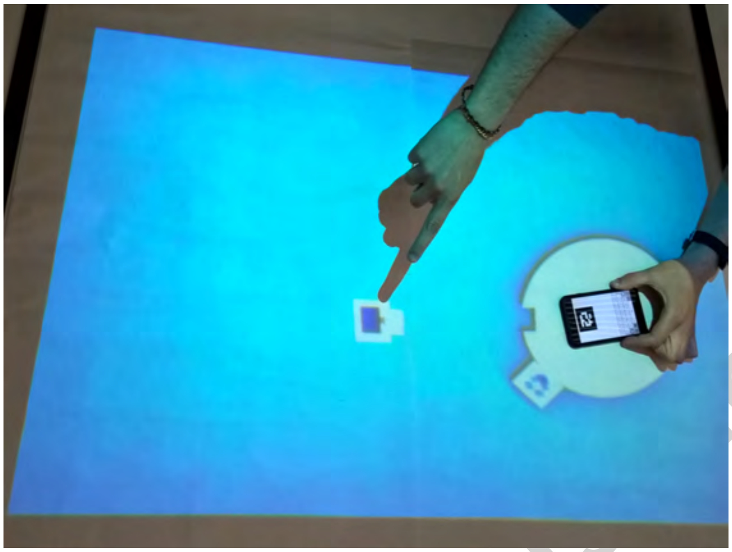


Figure 3: An example of a workflow being assembled using TAPAS: a keyboard widget is displayed on the smartphone once a new piece requiring an input is assembled.

Widgets vary depending on the type of input required: selecting a single option among several will prompt the user with a list box, a single action to be performed will display a button, and a generally unstructured raw text will present a keyboard (figure 3 and 4b). Once a user enters the requested input on a widget, the latter disappears from the smartphone and the projected halo surrounding it opens up a new hollow to allow for the next piece to be inserted (figure 4c); then using the input, only the hollow that is compatible with it is displayed, aiding users by preventing invalid compositions.

A puzzle piece instance can be added only to one user workflow, but it can be respawned by TAPAS later to make it available to other users; all communications through the smartphone and the display are managed via HTTP over the local Wi-Fi network, allowing for network outages.

The features currently available on our prototype, each rendered with a different puzzle piece, are: (1) selecting and downloading a file from the user's Dropbox account; (2) displaying a downloaded PDF file or an image on the main tabletop screen; (3) searching for a book in the university library and retrieving its location inside the building depicted in an image; and (4) sending a text document to a specified email address.

(a) The first piece is selected and added to the current workflow.

(b) The corresponding widget is displayed on the smartphone's display waiting for user input.

(c) Once the input is inserted, a piece whose input matches the current workflow's output can be added.

(d) Finally, the workflow is completed and the user can run it from her smartphone.

Figure 4: A step-by-step walkthrough of building a workflow with TAPAS.

For instance, one could pick 1 and 2 (in this order) and the composed application would download a PDF from the user's Dropbox folder and display its content on the big screen (as depicted in figure 4); composing 3 and 2 together would result in looking for an available book in the university library and displaying on the big screen a map depicting its location. These features have been designed with a specific scenario in mind, i.e. providing an interactive public display in an educational space to foster students' interaction on different projects; TAPAS has been designed with an open architecture (see figure 2) so that new services and corresponding puzzle pieces can easily be added depending on usage scenarios.

Summarizing, our prototype allows users to develop simple workflows while interacting with a TUI-based tabletop system installed in public spaces, thus empowering them to adapt and repurpose the latter to their needs.

4. Evaluation

We evaluated TAPAS twice: the first evaluation involved end users in a specific scenario, namely second year university students; they usually share resources with each other and gather information from public displays found within departments' foyers or the library in order to review lectures or complete their coursework. In particular, our participants — selected among Brunel University second year students in the Department of Computer Science, College of Engineering and Design — are required to collaborate on a project including many weekly meetings around shared spaces. This presented the right challenge for our application, as the public displays currently available offer services that are only partially relevant and highly scattered for the students' projects and might lead to their interest waning and the under utilisation of such expensive facilities.

Our study allowed us to investigate how TAPAS might be employed in such a real-world scenario, i.e. in-the-wild, but also to better define user requirements and ascertain whether they are fully or partially met by our system, informing the following stages of its design. The second evaluation involved a group of interaction designers and experts and focused on the interaction modality we are proposing with our prototype. The results of both our preliminary studies will be a helpful guide for the redesign of our prototype, even though a fully in-the-wild study is still needed to draw more definitive broad conclusions.

4.1. User Study

To get a better understanding of the scenarios where Pervasive Displays might be used, we carried out the first part of our study in a university setting, where many public interactive displays are already being deployed and used; these deployments are not usually effective or adaptable to the multitude of usage contexts they need to deal with and are also affected by the so-called Display Blindness effect [54], whereby they are usually overlooked due to people's low expectations of their content value.

4.1.1. Participants and procedure

We were interested in investigating the traditional usage contexts of a specific user group — namely Computer Science undergraduates during their second year — and how our prototype could help them; as part of their degree, students are clustered into groups of 4-6 and assigned with an Android application project to be undertaken during the course of the year, with the supervision of a teaching staff member, whom they usually meet all together as a group once a week.

Students are required to meet and work collaboratively every week, normally in the library or in one of the college's meeting rooms, and can use a range of available tools to work together and share information with each other (online dedicated forums or drives, laboratory spaces with coding facilities, etc.). The objective of these meetings is not to develop the Android application — which is an individual task — but to coordinate and organize a project plan, eventually designing a Gantt diagram with which students will split the workload into individual tasks. Our study has been conducted partially in-the-wild, since it took place in one of these facilities (a real world setting addressing real world problems) but with a researcher present (partially controlled).

The study involved three groups of students in their second year, made up respectively of four (1 female, 3 males), five (1 female, 4 males) and six (all males) students, reflecting the real project activity requirements and average group size; participants had no prior knowledge of TAPAS, but attended their introductory programming course during their first year, thus they already had some programming and problem solving experience. The study was conducted in three different sessions, one for each group; we conducted the study within the University facilities, in a room inside the Department of Computer Science designated to students and staff meetings. Each session lasted one hour and was made up of two consecutive activities (each half an hour long). The first activity addressed the scenario of group project meetings, their current practices and requirements for these. The second activity was a preliminary evaluation of our prototype's feature set and interaction modality. For the latter, we presented the students with TAPAS as a "provotype" — i.e., a provocative prototype, namely a prototype that deliberately challenges stakeholders' conceptions by reifying and exposing tensions of existing practice in a context of interest [55]; this includes a small set of features highly tailored to the evaluation scenario (i.e., university students collaborating with each other), which was the first step in proving our concept. The use of the "provotype" was meant to evaluate the current status of the application and especially to elicit the interaction modality requirements that might not have been easily gathered employing only a paper and pencil approach.

4.1.2. Elicitation of user activities

During the first activity we asked participants to tell us about the tasks, tools and public resources offered by the University that they would normally use during their weekly gatherings; we provided them with a non-exhaustive sample of icons representing some of the traditional resources and tools they might use, such as books, papers, search engines, smartphones, public display applications and so on. We asked them to place the relevant icons on a sheet of paper, which was divided into 3 different sections: before, during and after the meeting. Participants could use as many icons as they wanted, draw new ones, use post-it notes and link items together. In the end they had to produce an accurate picture of all the activities and tasks usually performed during a meeting and the kind of preparation each one of them requires, as well as all the further activities it might trigger; an example of the final result is depicted in figure 5.

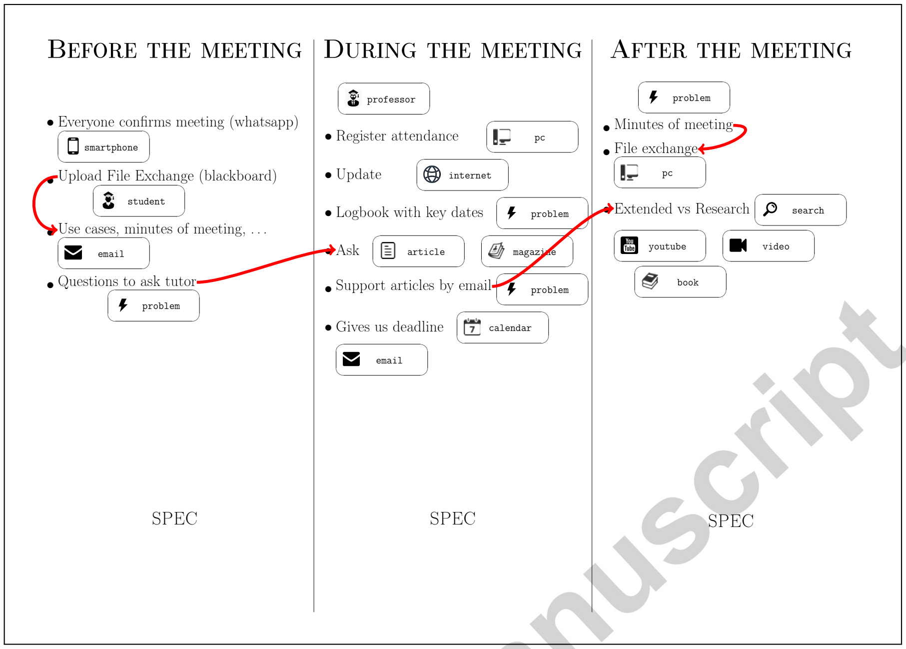


Figure 5: A snapshot of the rich picture generated by one group participating in our study.

The Rich Picture methodology was part of Checkland's Soft Systems Methodology to gather information about a complex situation [56]; Rich Pictures are used before clearly knowing what is to be considered a process and what a structure. They aim at representing a real situation with no constrained ideas. Due to its uncontrolled nature, this methodology is suitable to analyze our in-the-wild scenario, since it is often not easy to clearly separate the processes and structures involved.

Even though there is no specific notation for a Rich Picture and thus they can be misinterpreted, their informality helps communication with users, and might be coupled with an interview and the use of a prototype to allow users to be immersed in the scenario they are modeling [57]. Hence, while building the Rich Picture, we carried out a semi-structured interview in order to control misinterpretations; its results were clustered into *post hoc* generated categories [58]. We present the categories generated by the interviews in the following:

Scheduling activities. Students use an instant messaging tool to schedule meetings and discuss urgent matters with each other before a meeting, due to its dual real-time and asynchronous nature; they use the same tool to agree on issues to be brought to their supervisor's attention in the next meeting and build a collaborative agenda for it. During their meetings they review upcoming group and single member deadlines and milestones following their tutor's suggestions, storing their progress in each student's logbook, which contains the whole group's progress as well as each member's individual progress. Due to our previous knowledge of student activities our current prototype allows users to access a shared resource, such as their logbook, while giving each one of them a personalized view of their own progress. Nevertheless, from the semi-structured interviews it seems that our prototype will require some form of policy administration on shared-resource editing rights, which will definitely be considered as part of the next iteration of TAPAS.

Reporting activities. Each student's logbook also contains a report on the progress made so far; students describe how they have handled completed tasks and report problems they are encountering through the development process that will be then discussed with their tutor. Relevant resources such as papers or books suggested by their tutor or found by individual members are brought to the meeting and shared with the group as a whole or to subgroups (or even with single participants) depending on the scope of their different tasks. Usually only sharing requests are handled during a meeting, leaving actually sending out the resource to the right members as a post-meeting activity, which is subject to mistakes and forgetfulness. Our application allows sharing of resources from one member's private document library to others instantaneously, although thanks to requirements gathering we plan to include in future versions the ability to set groups of users as recipients.

Discussion activities. Discussions happen throughout all the three phases: before and after the meeting students use instant messaging tools to discuss pressing issues they came across during the development, or email for longer and more detailed discussions, seeking advice and suggestions from their peers. During the meeting itself the group discussion mainly focuses on issues relevant to all the members rather than individuals, but it may occasionally involve subgroups working on similar tasks. Using the large tabletop screen those requirements are naturally met by our prototype. Due to its collaborative features, it can be used to show all the other members some interesting resource and thus foster discussion among members of a groups. The prototype also makes it easy to hold multiple discussions between different subgroups.

4.1.3. Elicitation of interaction modalities

After the first activity (gathering requirements), we then proceeded with the second activity (30 minutes long) by briefly introducing the current version of TAPAS to participants, explaining to them how the system works. We then let them play with it for 15 minutes (figure 6), and finally carried out a semi-structured interview — mainly focused on the proposed interaction modality. We reminded them that our objective for this activity was to elicit the interaction modalities requirements that might not easily have been gathered just by employing a Rich Picture approach.

Results point out how TAPAS offers a quite satisfactory user experience; as expected, students' feedback mostly focused on missing features and the interaction with the system.

Each group managed to successfully assemble (at least once) two workflows: the first one started with downloading a PDF file from a Dropbox account and displaying a preview on the main tabletop surface, while the second one started with looking for a specific book in the university library and depicting its location on the main screen. One group even assembled a more complex workflow, consisting of the download of a text file from Dropbox and its subsequent dispatch via email to an address they chose. Indeed, all these three workflows might come in handy during a students' meeting, according to the Rich Picture's results: the first two workflows belong to the "Discussion activities", and the third one to the "Reporting activities".

From the feedback we have obtained it is clear how a Tangible User Interface (TUI) is an easy and effective way of interacting with the system throughout the composition of a workflow. Even though all our participants are Computer Science undergraduates, their second-year group project is their first chance of tackling a wider problem solving scenario, unlike their first year's individual development of smaller applications. This more complex project required them to learn abstraction and decomposition skills, whilst collaborating with peers. Using the puzzle metaphor and workflows together with tangible interaction seemed to help them build the required Computation Thinking skills: for instance, collaboratively planning and designing the application's tasks and assigning them to each participant seems like a suitable scenario to practice abstraction and composition skills. Moreover, as with API development, the recipe metaphor provides different levels of transparency and abstractions useful to generalize the problem whilst assembling a puzzle might help with decomposing a bigger problem into smaller ones [59].

Nonetheless, the feedback showed that tangible interaction is not very "natural" when it comes to manipulating their output: every participant trying out the prototype attempted to move images displayed on the screen with their fingers, suggesting that manipulating items through objects might feel "natural" only when operating in composition/developing mode, and not when there is actual content the user needs to directly manipulate available on the screen. This follows directly from our choice of employing a Programming by Instruction (PbI) paradigm, which uses a syntactic construct to specify a workflow's instructions as opposed to exploiting only contextual actions on resulting artifacts — i.e., Programming by Demonstration (PbD).

From the interaction point of view we noticed one interesting remark made by one of the participants: continuously tracking the smartphone's position on the surface using a fiducial marker requires the user to not cover its display with her hand when moving it; however, the user's hand position on the smartphone might depend on her posture: if the user is standing straight, it feels more "natural" to hold it from above — thus covering the fiducial marker with her palm — while if sitting down, the user might feel more comfortable grabbing it from the side, without covering its display, allowing its movements to be tracked. Because the majority of existing smartphones are shaped in the same way, it is worth studying this effect in more detail, in order to establish whether we could provide users with a physical enclosure affording the “right” way of holding the smartphone or whether it is a negligible effect when the system runs on horizontal displays of a certain distance from the floor.

Figure 6: One of the participating groups to our study working with TAPAS.

Summarizing, we gathered several detailed scenario requirements from users in the form of three usage contexts, which targeted scheduling, reporting and discussion activities; we highlighted how the current version of TAPAS deals with them and how we are going to address those that are not yet satisfied. The same users appear to cope easily with TAPAS’ interaction modality during the workflow editing phase, but we will need to devise a different interaction style when it comes to manipulating their results.

4.2. Designer Study

We also interviewed three interaction design experts to get feedback on the modality we have implemented in TAPAS; we carried out the interviews in a controlled environment (figure 7), namely during a workshop on the island of Tiree, during the bi-annual Tiree Tech Wave, a gathering of experts in various fields, ranging from interaction designers and artists to computer scientists. The study involved simultaneously two HCI experts and a product designer and lasted 45 minutes. We briefly introduced our prototype to them, explaining the rationale behind its design and the scenarios we are targeting; then we gave them a demonstration of how it works, going through some examples of its usage in a real world scenario. Finally we carried out a semi-structured interview focusing on the strengths and weaknesses of our prototype in relation to the interaction modality and its applicability in-the-wild, more precisely covering the easiness of the puzzle metaphor, the use of smartphones as tangible objects, possible application scenarios and future features.

Designers liked the overall idea and the personalization approach for Pervasive Display scenarios, namely using a smartphone as a tangible instead of just a passive object to identify users and link their personal information with the movements they perform on the very same device. In particular, they liked the puzzle metaphor since it looked a straightforward way of understanding the composition of workflows to address users' needs.

They recognized the potential of such a system in public spaces, due to its ease of deployment and the cheapness and high availability of the technologies involved: thanks to the simple architecture, TAPAS allows deployment in any digitally augmented surface just by installing an RGB camera and running the application on a production server; it can be left in public spaces for a long period of time without the need to perform mundane maintenance operations

Figure 7: The designer study setting.

aimed at adding new features, since users are empowered to repurpose it themselves.

Some of their suggestions focused on the way TAPAS presents data to users and the use of the dynamic widget to get some input from them: due to the kind of data handled right now — namely lists of files within directories or book titles in a database — it makes sense to prompt users to choose an option from a list or offer a keyboard to input raw text. Nevertheless, this will not be the case if we have to deal with more structured data types, such as points of interest on a map: therefore, they suggested that due to the complexity of workflows that might be put together by final users, widgets might be designed to be more flexible and personalizable depending on the two-fold level of interaction between the user perspective and data perspective related to the specific data handled by the widget. They emphasized that the two perspectives are interlinked and reinforced mutually. We propose to consider elements of human-centered information visualization in the redesign of the widgets for the next interaction prototype; for instance, by following visual metaphors that incorporate semantic relationships of visual objects both in the physical (tangible) and virtual (digital) world [60, 61].

Furthermore, interviewees pointed out how this continuous back and forth movement, between interacting with the smartphone to input data and with the large display to assemble workflows, might be confusing for users: interacting with two different devices, each one with a different interaction style — i.e. tangible on the tabletop, multi-touch on the smartphone — and different underlying metaphors, requires a relatively high cognitive effort in constantly switching paradigm and some users might also miss what is happening on one device while they are too focused on interacting with the other. That is why interviewees suggested keeping the tabletop as the main interaction focus by providing a mixed interaction modality: moving the smartphone will still be used to assemble the puzzle pieces but once one of them requires a certain input, the widget will appear on the tabletop surface — close to it — and the user will interact with it using her fingers.

The final observation concerns the puzzle metaphor we are using: although it appears to be quite an easy to grasp concept, we might need to offer some additional visual cues to improve its efficacy; interviewees suggested that in addition to shapes to indicate functions compatible with the currently generated output, we might highlight the available ones and darken the incompatible ones, even when the former are not available due to network outages or other problems; or associate colors to shapes.

Indeed, there are clearly positive elements in our design for End-User Development (EUD) of workflows to adapt public display services to users' needs, such as the puzzle metaphor, the use of the smartphone as being tangible and personal, and the ease of prototype deployment in-the-wild due to its low-cost and flexible architecture. Nonetheless, there are some major challenges to be addressed in future in terms of interaction design requirements, such as the flexibility/programmability of widgets and improving the puzzle metaphor to highlight available functionalities.

5. Discussion

From our study we identified two relevant challenges in the field of Tangible Programming on public interactive displays: the first stems from our user study with students and is about the duality of composing workflows and executing workflows in tangible environments; the second challenge has emerged during the study with designers and is related to the use of Visual Languages in the domain of Tangible Programming.

The user experience seems to differ when the tangible interaction is used for composing services with the puzzle metaphor (positive experience) from when they interact and collaborate on the results of the workflow execution through their smartphones (less positive experience). This could be due to the different set of constructs involved within each stage:

1. Building a workflow requires the user to deal with abstract concepts — like functions and constraints — that are not naturally coupled with any existing physical counterpart; providing users with an intuitive metaphor (the puzzle) and enabling them to interact with the system in a natural way (through a tangible) might be an effective strategy to help them build the right mental model, together with exposing the

right transparency level of the workflows' inner logic in order to improve abstraction and decomposition skills, indeed helping to develop their Computational Thinking abilities.

2. In a Natural User Interface (NUI) based environment, direct manipulation of contents is more intuitive than using intermediate control mechanisms; hence, when it comes to manipulating results produced by their workflows, users require the interface to be completely transparent, without any syntactical — least of all tangible — artifact to operate on an environment's constructs.

This contrast is also evident from the literature (see section 2.2) highlighting the many differences between the Programming by Demonstration (PbD) and Programming by Instruction (PbI) paradigms: due to its very nature, when a system exploits PbD, the composition and execution environments are perfectly overlapped, i.e. the same artifacts the users operate on to program the system are used also to interact with its results, as with Robot Programming by Demonstration; in Robot Programming by Demonstration users teach movements to a robot by simply simulating them directly onto its body. This is radically different from a PbI approach, where the two environments — composition and execution — are generally detached from one another, each one using different metaphors and concepts, e.g., in Yahoo! Pipes there is a visual editor for composing a pipe (data-flow) that generates a specific execution environment made of Graphical User Interface (GUI) elements as designed by the user. While this distinction might be overlooked from an interaction perspective when a system only relies on a GUI, it becomes more relevant when it is about Tangible User Interfaces (TUIs). Even though PbI seemed the right paradigm to choose in our scenario due to its generalizability and the benefits brought to Computational Thinking skills, we argue that choosing the right paradigm according to the naturalness of interaction is clearly scenario-dependent, as is often the case with Domain Specific Visual Languages.

From the second study with designers an interesting challenge has emerged which is related to the use of Visual Languages with TUIs. In particular, we have noted that the majority of examples we found in the literature (see section 2.2), including our prototype, use Visual Languages when employing a PbI paradigm.

Visual Languages have been widely used within the field of End-User Development (EUD) in order to ease the development process for end users; the interaction paradigm used for Visual Languages is GUI-based, whilst due to our scenario, i.e. Pervasive Displays, a more natural way of allowing EUD would be to support Tangible User Interaction. One challenge would be to study whether there is an EUD paradigm more suitable for TUI environments: this challenge would require understanding whether any of the available paradigms, e.g., PbI and PbD, are suitable for Tangible Programming or if — on the contrary — new paradigms need to be introduced. There is some evidence, as in Robot Programming by Demonstration for instance, that PbD is suitable for that specific scenario using Tangible Programming but, as often happens in the EUD community, the solution might be domain dependent.

A final remark concerns the problem we were investigating first, namely fostering the long-term appropriation of Pervasive Displays by enabling users to repurpose them through EUD: during our first study we collected and clustered the requirements of a typical scenario where Pervasive Displays could already be used, but — due to their maintenance issues and progressive loss of interest by users — are not yet widespread. Our analysis reported three types of activities that end users need to be able to carry easily out with a Pervasive Display in order to properly support user needs in the scenario we considered: (1) scheduling, (2) reporting, (3) and discussion activities.

While ours was indeed just a preliminary study on a specific application domain, we can certainly use its findings to highlight some of the issues preventing Pervasive Display deployment in-the-wild for long periods of time. Supporting collaboration is definitely a much needed feature, both peer-to-peer — that is where all participants have the same role within the group (e.g., discussion activities) — and chaired modes (e.g., reporting activities); discovering user roles is the cornerstone, and the use of smartphones can definitely come in handy [39]. Moreover, Pervasive Displays need to support users in individual activities as well (e.g., scheduling activities), enabling them to use their preferred tools while carefully considering the resulting privacy issues; indeed, our choice of employing smartphones as tangible probes in TAPAS was influenced by privacy concerns, allowing us to draw upon user data while keeping the user in control of what she wants to share and with whom. For this reason, we are currently working on the TAPAS' web app in order to develop a more sophisticated interface that enables users to effectively tweak their privacy settings and control which data TAPAS can have access to.

Finally, as we previously stated, it is undoubtedly worth pointing out the shortcomings of our studies: the limited number of components developed and deployed to the system prevented us from fully evaluating its usage in a real in-the-wild scenario, thus our findings cannot be properly generalized for many other contexts. Yet, since we employed TAPAS as a provotype — that is to challenge users by exposing tensions and thus to support design explorations [55] — observations related to the interactions users and designers carried out can give us a good insight into its real usage. Moreover, a fully in-the-wild study is needed to properly highlight how TAPAS relates to mundane Pervasive Displays activities.

6. Conclusion

A fairly recent trend in the Pervasive Displays research area is to design long-term in-the-wild deployments outside controlled laboratory settings and without any researcher supervision; nevertheless, these deployments present two main drawbacks: the first is the expensive setup and maintenance and the second is the progressive loss of interest shown by users, due to the lack of new features satisfying their shifting needs. A way of tackling this problem is to allow users to adapt the system themselves without the intervention of the site managers.

In this paper we introduced TAPAS, an application running on a Pervasive Display system, which allows users to adapt and repurpose the system using their smartphones combining a tangible and visual interaction. We have detailed its architecture and highlighted the advantages and rationale behind its design following a workshop with experts, making the case for the ease and convenience of its in-the-wild deployment.

We evaluated TAPAS by carrying out a two-phase study, the first phase involving end users in a specific scenario — second year undergraduates working in groups — and the second phase with interaction designers. From the first study's results, it seems that our prototype provides a positive user experience and could be used in a collaborative project scenario where people work together to tackle a complex problem; a potential side effect caused by employing our prototype might be a development of Computational Thinking skills, thanks to our design rationale. However, from our findings it also appears that coupling tangible interaction with a Programming by Instruction paradigm causes an incompatibility of interaction styles between the composition and the execution environments, where the use of a different tangible-based syntactic construct in the former causes the need for a different interaction style to be used in the latter.

The second study we conducted to evaluate our prototype was focused on its interaction modality and involved a group of interaction design experts; the results show that participants liked the proposed interaction style, recognizing the potential of the exploited puzzle metaphor in easing the adaptation tasks for the end users. They also suggested extending the platform in order to cope with more complex data to be manipulated by end users. However, from the results it seems that exploiting Visual Languages within a Tangible User Interface system might not be the best way of providing users with a natural interaction experience, thus further investigations are needed to determine the role of the scenario in the choice of the right paradigm (i.e. Programming by Instruction or Programming by Demonstration).

In the future we plan to study in more detail issues arising from our findings, with particular attention to the main challenges discussed in section 5. We plan to exploit the feedback obtained from our studies in the next iteration of TAPAS' design and carry out additional evaluation studies in other public scenarios, such as university settings or urban areas, and in non-public collaborative contexts too, e.g., within a company. Moreover, further studies will be carried out in order to draw more definitive conclusions regarding the effect of the proposed interaction modality on the development of Computational Thinking skills, as well as within a fully in-the-wild setting, where participants will be prompted to use the system without any researchers' intervention. We also plan on studying whether extending TAPAS' functionalities without support for more complex workflows, as suggested by designers and users, might improve its adoption.

Acknowledgment

We kindly thank all Tiree Tech Wave participants and the students involved in our study for their commitment to this research. We would also like to thank all the members of the Human Centred Design Institute for their helpful support, especially Voula Gkatzidou, who provided valuable insights for the undertaking of this research.

References

[1] T. Turchi, A. Malizia, A. Dix, Fostering the adoption of Pervasive Displays in public spaces using tangible End-User Programming, in: 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), IEEE, 2015, pp. 37–45.

[2] H. Ishii, B. Ullmer, Tangible bits, in: the SIGCHI Conference, ACM Press, New York, New York, USA, 1997, pp. 234–241.

[3] T. S. Mcnerney, From turtles to Tangible Programming Bricks: Explorations in physical language design, Personal and Ubiquitous Computing 8 (2004) 326–337.

[4] M. Horn, E. T. Solovey, J. Crouser, R. Jacob, Comparing the use of tangible and graphical programming languages for informal science education, in: the SIGCHI Conference, ACM Press, New York, New York, USA, 2009, p. 975.

[5] D. Wang, T. Wang, Z. Liu, A tangible programming tool for children to cultivate computational thinking, The Scientific World Journal 2014 (2014) 1–10.

[6] D. Wigdor, D. Wixon, The Spatial NUI, in: Brave NUI World, Elsevier, 2011, pp. 33–36.

[7] L. Terrenghi, A. Quigley, A. Dix, A taxonomy for and analysis of multi-person-display ecosystems, in: Personal and Ubiquitous Computing, Vodafone Group Services R and D, Munich, Germany, pp. 583–598.

[8] A. Bellucci, A. Malizia, I. Aedo, Light on horizontal interactive surfaces: Input space for tabletop computing, ACM Computing Surveys 46 (2014) 1–42.

[9] C. Ardito, P. Buono, M. F. Costabile, G. Desolda, Interaction with large displays: A survey, ACM Computing Surveys 47 (2015) 1–38.

[10] A. Crabtree, A. Chamberlain, R. E. Grinter, M. Jones, T. Rodden, Y. Rogers, Introduction to the special issue of "The turn to the wild", ACM Transactions on Computer-Human Interaction 20 (2013).

[11] S. Hosio, J. Goncalves, H. Kukka, A. Chamberlain, A. Malizia, What's in it for me: Exploring the real-world value proposition of pervasive displays, in: PerDis 2014 - Proceedings: 3rd ACM International Symposium on Pervasive Displays 2014, Oulun Yliopisto, Oulu, Finland, pp. 174–179.

[12] N. Memarovic, M. Langheinrich, K. Cheverst, N. Taylor, F. Alt, P-LAYERS - A layered framework addressing the multifaceted issues facing community-supporting public display deployments, ACM Transactions on Computer-Human Interaction 20 (2013) 1–34.

[13] T. Turchi, A. Malizia, Pervasive displays in the wild: Employing end user programming in adaption and re-purposing, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Brunel University London, Uxbridge, United Kingdom, pp. 223–229.

[14] R. Jacob, A. Girouard, L. Hirshfield, M. Horn, O. Shaer, E. T. Solovey, J. Zigelbaum, Reality-based interaction, in: Proceeding of the twenty-sixth annual CHI conference, ACM Press, New York, New York, USA, 2008, p. 201.

[15] A. Manches, C. O'Malley, S. Benford, Physical manipulation, in: the

3rd International Conference, ACM Press, New York, New York, USA, 2009, p. 77.

[16] A. J. Parkes, H. S. Raffle, H. Ishii, Topobo in the wild, in: Proceeding of the twenty-sixth annual CHI conference, ACM Press, New York, New York, USA, 2008, p. 1129.

[17] M. Weiss, J. Wagner, Y. Jansen, R. Jennings, R. Khoshabeh, J. D. Hollan, J. Borchers, SLAP widgets, in: the SIGCHI Conference, ACM Press, New York, New York, USA, 2009, p. 481.

[18] J. Müller, T. Schwarz, S. Butscher, H. Reiterer, Back to tangibility, in: the 2014 International Working Conference, ACM Press, New York, New York, USA, 2014, pp. 57–64.

[19] S. Subramanian, D. Pinelle, J. Korst, V. Buil, Tabletop Collaboration through Tangible Interactions, in: 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2007), IEEE, 2007, pp. 412–417.

[20] M. Hancock, O. Hilliges, C. Collins, D. Baur, S. Carpendale, Exploring tangible and direct touch interfaces for manipulating 2D and 3D information on a digital table, in: the ACM International Conference, ACM Press, New York, New York, USA, 2009, p. 77.

[21] D. Kirk, A. Sellen, S. Taylor, N. Villar, S. Izadi, Putting the physical into the digital: Issues in designing hybrid interactive surfaces, in: People and Computers XXIII Celebrating People and Technology - Proceed-

ings of HCI 2009, Microsoft Research Cambridge, Cambridge, United Kingdom, pp. 35–44.

[22] P. Marshall, Do tangible interfaces enhance learning?, in: the 1st international conference, ACM Press, New York, New York, USA, 2007, p. 163.

[23] A. Antle, A. F. Wise, Getting down to details: Using theories of cognition and learning to inform tangible user interface design, Interacting with Computers 25 (2013) 1–20.

[24] M. Horn, J. Crouser, M. U. Bers, Tangible interaction and learning: The case for a hybrid approach, Personal and Ubiquitous Computing 16 (2012) 379–389.

[25] J. M. Wing, Computational thinking, Communications of the ACM 49 (2006) 33.

[26] Foreword, in: 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond), IEEE, 2015, pp. vii–viii.

[27] N. Tillmann, M. Moskal, J. De Halleux, M. Fahndrich, S. Burckhardt, Touchdevelop: app development on mobile devices, in: Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, ACM, p. 39.

[28] S. Cuccurullo, R. Francese, M. Risi, G. Tortora, MicroApps Development on Mobile Phones, 2011.

[29] H. Lieberman, Programming by example (introduction), Communications of the ACM 43 (2000) 72–74.

[30] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Robot Programming by Demonstration, in: Springer Handbook of Robotics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 1371–1394.

[31] J. Lee, L. Garduño, E. Walker, W. Burleson, A tangible programming tool for creation of context-aware applications, in: the 2013 ACM international joint conference, ACM Press, New York, New York, USA, 2013, p. 391.

[32] E. Mugellini, D. Lalanne, B. Dumas, F. Evéquoz, S. Gerardi, A. Le Calvé, A. Boder, R. Ingold, O. Abou Khaled, MEMODULES as Tangible Shortcuts to Multimedia Information, in: Virtual and Mixed Reality, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 103–132.

[33] D. Wang, Y. Zhang, T. Gu, L. He, H. Wang, E-Block, in: Adjunct proceedings of the 25th annual ACM symposium, ACM Press, New York, New York, USA, 2012, p. 71.

[34] A. Sipitakiat, N. Nusen, Robo-Blocks, in: the 11th International Conference, ACM Press, New York, New York, USA, 2012, p. 98.

[35] R. Sharp, K. Rehman, The 2005 UbiApp workshop: What makes good application-led research?, in: IEEE Pervasive Computing, Intel Research Cambridge Lab, Cambridge, United Kingdom, pp. 80–82.

[36] N. Davies, M. Langheinrich, S. Clinch, I. Elhart, A. Friday, T. Kubitza, B. Surajbali, Personalisation and privacy in future pervasive display networks, in: the 32nd annual ACM conference, ACM Press, New York, New York, USA, 2014, pp. 2357–2366.

[37] A. Bellucci, A. Malizia, P. Díaz, I. Aedo, Don't Touch Me: Multi-user annotations on a map in large display environments, in: Proceedings of the Workshop on Advanced Visual Interfaces AVI, Universidad Carlos III de Madrid, Madrid, Spain, pp. 391–392.

[38] P. Cremonesi, A. Di Rienzo, C. Frà, F. Garzotto, L. Oliveto, M. Valla, Personalized interaction on large displays, in: the 2014 International Working Conference, ACM Press, New York, New York, USA, 2014, pp. 353–354.

[39] S. Clinch, Smartphones and pervasive public displays, IEEE Pervasive Computing 12 (2013) 92–95.

[40] T. Kubitza, S. Clinch, N. Davies, M. Langheinrich, Using mobile devices to personalize pervasive displays, ACM SIGMOBILE Mobile Computing and Communications Review 16 (2013) 26.

[41] S. Izadi, H. Brignull, T. Rodden, Y. Rogers, M. Underwood, Dynamo, in: the 16th annual ACM symposium, ACM Press, New York, New York, USA, 2003, pp. 159–168.

[42] A. Dix, C. Sas, Mobile personal devices meet situated public displays: Synergies and opportunities, International Journal of Ubiquitous Computing (2010).

[43] R. Hardy, E. Rukzio, M. Wagner, M. Paolucci, Exploring expressive nfc-based mobile phone interaction with large dynamic displays, in: Near Field Communication, 2009. NFC '09. First International Workshop on, pp. 36–41.

[44] P. Dalsgaard, K. Halskov, Tangible 3D tabletops, interactions 21 (2014) 42–47.

[45] C. Ardito, M. Francesca Costabile, G. Desolda, R. Lanzilotti, M. Matera, A. Piccinno, M. Picozzi, User-driven visual composition of service-based interactive spaces, Journal of Visual Languages & Computing 25 (2014) 278–296.

[46] S. N. H. Mohamad, A. Patel, R. Latih, Q. Qassim, L. Na, Y. Tew, Block-based programming approach: Challenges and benefits, in: Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, ICEEI 2011, Universiti Kebangsaan Malaysia, Bangi, Malaysia, IEEE, 2011, pp. 1–5.

[47] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, Y. Kafai, Scratch: Programming for all, Communications of the ACM 52 (2009) 60–67.

[48] A. Malizia, A. Bellucci, P. Díaz, I. Aedo, S. Levialdi, EStorys: A visual storyboard system supporting back-channel communication for emergencies, Journal of Visual Languages & Computing 22 (2011) 150–169.

[49] V. Maquil, E. Ras, O. Zephir, Understanding the Characteristics of Metaphors in Tangible User Interfaces., Mensch & Computer Workshopband (2011).

[50] D. Wigdor, D. Wixon, No Touch Left Behind, in: The Spatial NUI, Elsevier, 2011, pp. 81–95.

[51] Q. Bonnard, S. Lemaignan, G. Zufferey, A. Mazzei, Chilitags 2: Robust fiducial markers for augmented reality and robotics, 2013.

[52] M. Kaltenbrunner, T. Bovermann, R. Bencina, E. Costanza, TUIO: A protocol for table-top tangible user interfaces, in: Proc. of the The 6th Int'l Workshop on Gesture in Human-Computer Interaction and Simulation, pp. 1–5.

[53] J. Danado, F. Paternò, Puzzle: A Visual-Based Environment for End User Development in Touch-Based Mobile Phones, in: Virtual and Mixed Reality, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 199–216.

[54] J. Müller, D. Wilmsmann, J. Exeler, M. Buzeck, A. Schmidt, T. Jay, A. Krüger, Display Blindness: The Effect of Expectations on Attention towards Digital Signage, in: Virtual and Mixed Reality, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 1–8.

[55] L. Boer, J. Donovan, Provotypes for participatory innovation, in: the Designing Interactive Systems Conference, ACM Press, New York, New York, USA, 2012, p. 388.

[56] F. P. Wheeler, P. Checkland, Systems Thinking, Systems Practice: Includes a 30-Year Retrospective, The Journal of the Operational Research Society 51 (2000) 647.

[57] A. Sutcliffe, Scenario-Based Requirements Engineering (SCRAM), in: User-Centred Requirements Engineering, Springer London, London, 2002, pp. 127–147.

[58] N. R. Pandit, The creation of theory: A recent application of the grounded theory method, The qualitative report (1996).

[59] J. M. Wing, Computational thinking, in: 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), IEEE, 2011, pp. 3–3.

[60] A. Majumder, B. Sajadi, Large area displays: The changing face of visualization, Computer 46 (2013) 26–33.

[61] A. Bigelow, S. Drucker, D. Fisher, M. Meyer, Reflections on how designers design with data, in: the 2014 International Working Conference, ACM Press, New York, New York, USA, 2014, pp. 17–24.