

Citation for published version:

Tommaso Turchi, Alessio Malizia, and Alan Dix, 'TAPAS: A tangible End-User Development tool supporting the repurposing of Pervasive Displays', *Journal of Visual Languages & Computing*, Vol. 39: 66-77, April 2017.

DOI:

<https://doi.org/10.1016/j.jvlc.2016.11.002>

Document Version:

This is the Accepted Manuscript version.

The version in the University of Hertfordshire Research Archive may differ from the final published version.

Copyright and Reuse:

© 2016 Elsevier Ltd.

This manuscript version is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License CC BY NC-ND 4.0

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way.

Enquiries

If you believe this document infringes copyright, please contact the Research & Scholarly Communications Team at rsc@herts.ac.uk

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 [13]. 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

17 a physical shape [1]. By taking advantage of our innate dexterity for ob-
18 ject manipulation, TUIs have proven to be very effective in making highly
19 abstract activities such as programming more direct and accessible. In ad-
20 dition, there are interesting preliminary results linking the usage of tangible
21 tools with increased ability to model abstract concepts [2, 3]; these findings
22 suggest that physical manipulation acts as a scaffold between the real and
23 digital, enhancing Computational Thinking skills [4].

24 Nevertheless, it is not just the input modality that makes an interface
25 natural: it has to leverage each user’s capabilities and meet her needs while
26 fitting the current task and context demands. The design of innovative digital
27 systems like Pervasive Displays has indeed followed many principles [5] with
28 the aim of making interactions as *natural* as possible, in order to support their
29 appropriation and widespread use; these systems are composed of various-
30 sized displays supporting a many-to-many interaction modality and allowing
31 “many people to interact with the same public screens simultaneously” [6].

32 In recent years, thanks to the newly available technologies and the intu-
33 itive interaction capabilities of Pervasive Displays, they have spread around
34 public areas such as museums, tourist information centers, universities, shop-
35 ping malls and various other urban locations [7, 8]. A new trend has recently
36 emerged within Pervasive Displays research, namely to design large and long-
37 term deployments outside traditional controlled laboratory settings with no
38 researchers’ supervision, in other words *in-the-wild*. These studies evaluate
39 artifacts in their habitual use context within people’s lives; this means ob-
40 serving and recording what people do with them and how their usage changes
41 over time [9].

42 Yet, long term deployments of Pervasive Displays present two main draw-
43 backs [10]: (1) their setup and daily operational activities are expensive, and
44 (2) users and site managers tend to lose interest in their usage and mainte-
45 nance over time. Even if at first it is easy to advertise the provided benefits
46 through articles and papers presenting similar success stories, problems start
47 to surface when the initial buzz and enthusiasm (novelty) wears off and man-
48 agers have to carry out the daily maintenance tasks. In addition, keeping
49 these systems interesting over time by constantly providing them with fresh
50 content has proven to be particularly challenging [11].

51 To solve these problems, Hosio et al. [10] suggest that allowing a degree of
52 appropriation when designing Pervasive Displays might enable all the stake-
53 holders to understand how such systems could relate to the ordinary activities
54 they often take for granted, leading to a more sustained and prolonged use.
55 Moreover, their public and moderated nature does not allow the provision of
56 a broad set of general purpose and unfixed features, because users' interests
57 and needs are commonly heterogeneous and continuously evolving. Thus,
58 Pervasive Displays need to be adapted to all the different users' needs and
59 repurposed as those needs shift over time to promote a more serendipitous
60 and prolonged usage.

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

67 of a TUI to carry out their repurposing.

68 The contributions of this paper are twofold. First, we present an ap-
69 plication for Pervasive Displays, combining a TUI — employing the user’s
70 smartphone as the physical probe — with a visual interface projected onto
71 a tabletop display; our prototype — called TAngible Programmable Aug-
72 mented Surface (TAPAS) [12, 13] — aims at providing users with an easy
73 and simple way of composing simple task-oriented applications (e.g., down-
74 loading a PDF from the user’s Dropbox account and displaying its preview
75 on the main tabletop screen). Second, we highlight some of the main chal-
76 lenges faced by Tangible Programming on Pervasive Displays stemming from
77 two preliminary studies we carried out with end users and designers.

78 In particular, we outline the process we went through in designing TAPAS,
79 whose interaction paradigm stems from the results of a workshop we carried
80 out with expert designers, which we used to collect insightful ideas and de-
81 sign challenges related to the introduction of an EUD metaphor to a tangible
82 interactive tabletop. Our application is designed to foster collaboration and
83 support appropriation of Pervasive Displays systems in many different con-
84 texts of use (e.g., within a company to support users creating and sharing
85 data analyses); the first evaluation scenario we have selected is within an
86 educational space to foster students’ collaboration on different projects dur-
87 ing their recurring group meetings. To validate the efficacy of the proposed
88 interaction for guiding users in the composition of different applications, we
89 carried out two preliminary formative evaluations within a collaborative work
90 scenario, involving, respectively, second year university students working on
91 a group project and interaction designers. Strictly speaking, since this study

92 is not completely in-the-wild, the findings can only be leveraged to improve
93 the design of TAPAS, thus further studies are needed to draw more definitive
94 conclusions over the proposed interaction modality within this scenario.

95 **2. Related Works**

96 *2.1. Tangible User Interfaces*

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

103 The term ‘natural’ has been used in a rather loose fashion, meaning in-
104 tuitive, easy to use or easy to learn; many studies argue that we can design
105 a natural interaction either by mimicking aspects of the real world [14] or
106 by drawing on our existing capabilities in the communicative or gesticulative
107 areas [5].

108 One of the most successful and developed approaches falling into the first
109 category has been introduced by Ishii et al. [1] and is known as Tangible
110 User Interfaces (TUIs). The aim of TUIs is to give bits a directly accessible
111 and manipulable interface by employing the real world, both as a medium
112 and as a display for manipulation; indeed by connecting data with physical
113 artifacts and surfaces we can make bits tangible.

114 Many studies in this research area investigate the supposed benefits of-
115 fered by this interaction paradigm, ranging from intuitiveness [1], experiential

116 learning through direct manipulation [15, 16], motor memory [17], accuracy
117 [18], and collaboration [19]. Furthermore, the effects of employing a TUI to
118 interact with a digital system are certainly dependent on the tasks and do-
119 main, as many comparative studies suggest [17, 18, 20]; for this reason, Kirk
120 et al. [21] made the case for hybrid surfaces, employing physical elements
121 together with digital ones.

122 Researchers are also debating how employing TUIs reflects on learning
123 [3, 22, 23], with specific reference to highly abstract concepts: this stems
124 from Piagetian theories supporting the development of thinking — particu-
125 larly in young children — through manipulation of concrete physical objects.
126 Other studies [4, 24] are even linking this effect to the development of Com-
127 putational Thinking skills [25], namely a new kind of analytical thinking
128 integral to solving complex problems using core computer scientists’ tools,
129 such as abstraction and decomposition.

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

134 *2.2. End-User Development*

135 The End-User Development (EUD) research community always strives
136 to make programming tasks easier for end users (any computer user), thus
137 allowing them to adapt software systems to their particular needs at hand.
138 Visual Programming (VP) is one of the most well-studied techniques, aiming
139 at lowering barriers and often used to first introduce to coding: it reduces
140 the traditional syntactic burden of a programming language (often domain-

141 specific — i.e., tailored to a given application domain) by encapsulating it
142 with a visual representation of its instructions, using graphical tweaks to
143 communicate the underlying semantic rules at a glance.

144 Programming components can, for example, be represented by different
145 blocks, allowing users to combine them together to build a working program.
146 Constraints over different data types can be enforced by using different shapes
147 and allowing only matching inputs and outputs to be combined. Using blocks
148 to represent program syntax trees is a recent trend in designing VP systems
149 [26], as witnessed by the spread of Block Programming Environments like
150 Scratch¹, Microsoft Touch Develop [27], App Inventor², and MicroApp [28],
151 to name but a few.

152 Yet another way of aiding users in their programming task is by employ-
153 ing tangible objects. The existing literature can be clustered in two main
154 categories according to the paradigm employed: Programming by Demon-
155 stration (PbD) or Programming by Instruction (PbI). PbD, also known as
156 Programming by Example, enables users to teach new behaviors to the sys-
157 tem by demonstrating actions on concrete examples [29]. PbI, known as
158 Tangible (sometimes Physical) Programming within the TUI domain, takes
159 a traditional approach to programming, that is requiring users to learn and
160 employ a syntactic construct (e.g., text instructions, natural or visual lan-
161 guages) to impart instructions to the system.

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

¹<https://scratch.mit.edu>

²<http://appinventor.mit.edu>

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

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

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

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

197 *2.3. Pervasive Displays*

198 In the last two decades Pervasive Displays — namely ecosystems of various-
199 sized displays supporting simultaneous interactions with the same public
200 screens [6] — have become common within public areas. As well as at-
201 tracting the general public’s interest, this has led to the flourishing of an
202 active research community³. Most of the studies within this area are carried
203 out in-the-wild [35], in other words outside of controlled settings, in places
204 where such systems are commonly to be found.

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

³<http://pervasivedisplays.org>

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

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

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

237 interactions on the main tabletop.

238 Our system’s design exploits smartphones both as personalization trig-
239 gers and as interaction control mechanisms, in order to leverage the benefits
240 brought in by having a tangible interaction and an adaptable Pervasive Dis-
241 play.

242 **3. TAngible Programmable Augmented Surface**

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

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

253 *3.1. Design*

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

259 TAPAS' programming environment uses a Block-based Programming ap-
260 proach [46] — widely used by systems like Scratch [47] and Blockly⁴ — that
261 has proven to have a low learning threshold for non-programmers.

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

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

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

⁴<https://developers.google.com/blockly>

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

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

⁵<http://ifttt.com>

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

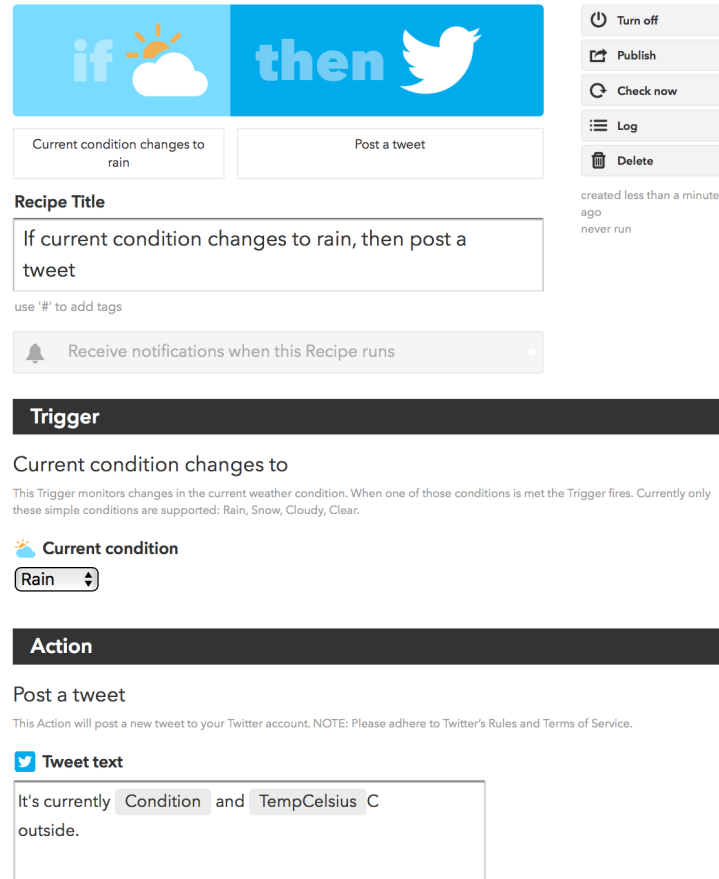


Figure 1: An example of a workflow created using IFTTT: when the condition in the user’s location changes to rain (trigger) it will automatically post a tweet (action).

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

311 After the introduction, participants started a 30-minute discussion about

312 ideas and challenges for the design of TAPAS' syntax, focusing on a collabora-
313 tive work scenario involving users with no previous programming experience.

314 *3.1.1. Preliminary Findings*

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

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

332 *Digital features.* Another set of suggestions were directed towards the digital
333 representation of our platform's syntax. First, the blocks' digital representa-
334 tion could help users understand components' constraints by using, respec-
335 tively, different and similar colors or shapes for incompatible and compatible

336 components. Also, since a workflow’s composition is usually performed one
337 component at a time, i.e. by selecting a function that will follow the latest
338 assembled one, our system might aid users on the next available components
339 to be chosen by changing the color or the shape of the currently assembled
340 workflow. Lastly, since TAPAS shows all available components at once, this
341 gives the user an overall view of the system’s capabilities. However, this also
342 allows users to make mistakes. TAPAS is intended to be used by inexperi-
343 enced users, so we need to assist users in finding the right way of assembling
344 different components, when they cannot figure it out themselves; a useful
345 suggestion in this regard is to provide some sort of “translation tool”, which
346 — once a user selects two blocks incompatible with each other — shows them
347 at least one possible way of choosing other components in between to connect
348 the two blocks, assisting users during the composition phase.

349 After collecting these suggestions from the workshop, we designed TAPAS
350 trying to fulfill the majority of them; we present the details of its implemen-
351 tation in the following section.

352 *3.2. Architecture*

353 TAPAS comprises a horizontal tabletop display and an RGB camera cap-
354 turing the movements of the users’ smartphones on the main display’s sur-
355 face using fiducial markers [51] (i.e. images used as a point of reference when
356 placed in the camera’s field of view), as summarized in figure 2; it supports
357 the Tangible User Interface Objects (TUIO) protocol [52], already adopted
358 by many research communities within the TUI area as a general and versa-
359 tile communication interface between tangible tabletop controller interfaces
360 and underlying application layers, which has been designed specifically for

361 interactive multi-touch tabletop surfaces.

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

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

383 *3.3. Interaction Paradigm*

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

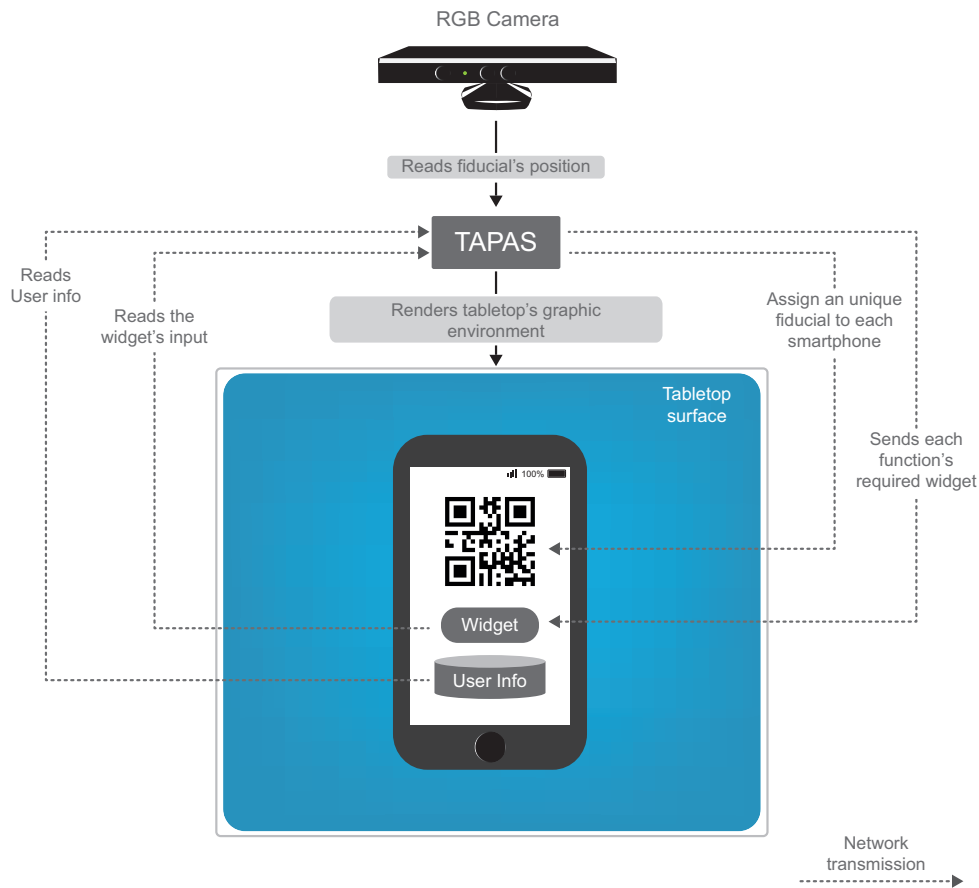


Figure 2: The architecture of TAPAS: using a fiducial marker — assigned by the application itself — and a RGB camera, TAPAS can track a smartphone’s movements on a tabletop surface; through the smartphone, TAPAS is able to link each and every smartphone’s movements to its users and display a corresponding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

386 is composed of different functions — or ingredients; moreover, a recipe can
 387 become a service itself, thus it can then be included in other recipes, fostering
 388 their reuse. In the future, users will be able to share their recipes or modify
 389 the ones they or others have created, just as they do with real recipes in
 390 their cookbooks. Thanks to the introduction of this recipe mechanism our

391 prototype allows users to share services with others who might have the same
392 needs. Furthermore, as would happen in real life, if someone does not have
393 a specific ingredient for a recipe she would seldom change recipe but instead
394 find a way of replacing an ingredient with one that is available, in agreement
395 with the results of the design workshop, which suggested providing some
396 sort of “translation tool” to help users finding missing components needed to
397 join two blocks. Moreover, if, for example, a service is not available due to
398 network failure, our recipe metaphor and the use of a smartphone still allows
399 data stored locally on the device to be used in services included in the recipe.

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

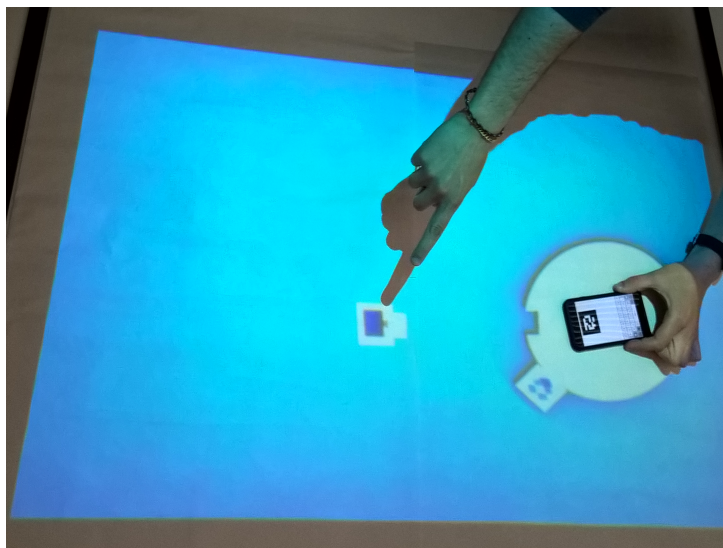


Figure 3: An example of a workflow being assembled using TAPAS: a keyboard widget is displayed on the smartphone once a new piece requiring an input is assembled.

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

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

428 The features currently available on our prototype, each rendered with

429 a different puzzle piece, are: (1) selecting and downloading a file from the
430 user's Dropbox account; (2) displaying a downloaded PDF file or an image on
431 the main tabletop screen; (3) searching for a book in the university library
432 and retrieving its location inside the building depicted in an image; and
433 (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.

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

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

447 **4. Evaluation**

448 We evaluated TAPAS twice: the first evaluation involved end users in a
449 specific scenario, namely second year university students; they usually share
450 resources with each other and gather information from public displays found
451 within departments’ foyers or the library in order to review lectures or com-
452 plete their coursework. In particular, our participants — selected among
453 Brunel University second year students in the Department of Computer Sci-
454 ence, College of Engineering and Design — are required to collaborate on
455 a project including many weekly meetings around shared spaces. This pre-
456 sented the right challenge for our application, as the public displays currently
457 available offer services that are only partially relevant and highly scattered

458 for the students' projects and might lead to their interest waning and the
459 under utilisation of such expensive facilities.

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

469 *4.1. User Study*

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

477 *4.1.1. Participants and procedure*

478 We were interested in investigating the traditional usage contexts of a
479 specific user group — namely Computer Science undergraduates during their
480 second year — and how our prototype could help them; as part of their
481 degree, students are clustered into groups of 4-6 and assigned with an Android

482 application project to be undertaken during the course of the year, with the
483 supervision of a teaching staff member, whom they usually meet all together
484 as a group once a week.

485 Students are required to meet and work collaboratively every week, nor-
486 mally in the library or in one of the college's meeting rooms, and can use
487 a range of available tools to work together and share information with each
488 other (online dedicated forums or drives, laboratory spaces with coding fa-
489 cilities, etc.). The objective of these meetings is not to develop the Android
490 application — which is an individual task — but to coordinate and organize
491 a project plan, eventually designing a Gantt diagram with which students
492 will split the workload into individual tasks. Our study has been conducted
493 partially in-the-wild, since it took place in one of these facilities (a real world
494 setting addressing real world problems) but with a researcher present (par-
495 tially controlled).

496 The study involved three groups of students in their second year, made up
497 respectively of four (1 female, 3 males), five (1 female, 4 males) and six (all
498 males) students, reflecting the real project activity requirements and average
499 group size; participants had no prior knowledge of TAPAS, but attended
500 their introductory programming course during their first year, thus they al-
501 ready had some programming and problem solving experience. The study
502 was conducted in three different sessions, one for each group; we conducted
503 the study within the University facilities, in a room inside the Department
504 of Computer Science designated to students and staff meetings. Each session
505 lasted one hour and was made up of two consecutive activities (each half an
506 hour long). The first activity addressed the scenario of group project meet-

507 ings, their current practices and requirements for these. The second activity
508 was a preliminary evaluation of our prototype’s feature set and interaction
509 modality. For the latter, we presented the students with TAPAS as a “provo-
510 type” — i.e., a provocative prototype, namely a prototype that deliberately
511 challenges stakeholders’ conceptions by reifying and exposing tensions of ex-
512 isting practice in a context of interest [55]; this includes a small set of features
513 highly tailored to the evaluation scenario (i.e., university students collaborat-
514 ing with each other), which was the first step in proving our concept. The use
515 of the “provotype” was meant to evaluate the current status of the applica-
516 tion and especially to elicit the interaction modality requirements that might
517 not have been easily gathered employing only a paper and pencil approach.

518 *4.1.2. Elicitation of user activities*

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

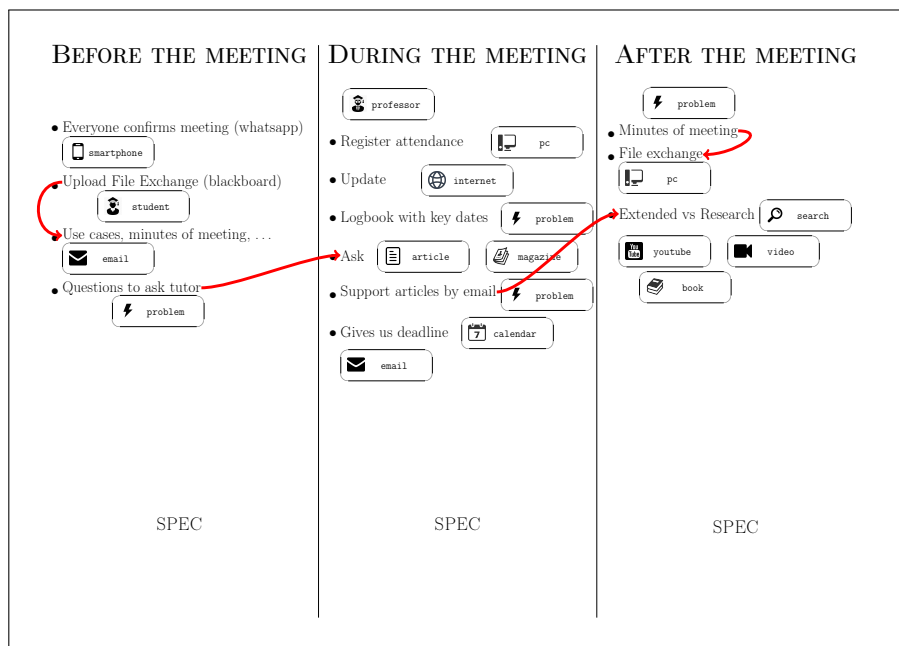


Figure 5: A snapshot of the rich picture generated by one group participating in our study.

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

539 Even though there is no specific notation for a Rich Picture and thus they
 540 can be misinterpreted, their informality helps communication with users, and
 541 might be coupled with an interview and the use of a prototype to allow users
 542 to be immersed in the scenario they are modeling [57]. Hence, while building
 543 the Rich Picture, we carried out a semi-structured interview in order to

544 control misinterpretations; its results were clustered into *post hoc* generated
545 categories [58]. We present the categories generated by the interviews in the
546 following:

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

562 *Reporting activities.* Each student’s logbook also contains a report on the
563 progress made so far; students describe how they have handled completed
564 tasks and report problems they are encountering through the development
565 process that will be then discussed with their tutor. Relevant resources
566 such as papers or books suggested by their tutor or found by individual
567 members are brought to the meeting and shared with the group as a whole
568 or to subgroups (or even with single participants) depending on the scope

569 of their different tasks. Usually only sharing requests are handled during a
570 meeting, leaving actually sending out the resource to the right members as
571 a post-meeting activity, which is subject to mistakes and forgetfulness. Our
572 application allows sharing of resources from one member's private document
573 library to others instantaneously, although thanks to requirements gathering
574 we plan to include in future versions the ability to set groups of users as
575 recipients.

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

588 *4.1.3. Elicitation of interaction modalities*

589 After the first activity (gathering requirements), we then proceeded with
590 the second activity (30 minutes long) by briefly introducing the current ver-
591 sion of TAPAS to participants, explaining to them how the system works.
592 We then let them play with it for 15 minutes (figure 6), and finally carried

593 out a semi-structured interview — mainly focused on the proposed interac-
594 tion modality. We reminded them that our objective for this activity was
595 to elicit the interaction modalities requirements that might not easily have
596 been gathered just by employing a Rich Picture approach.

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

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

610 From the feedback we have obtained it is clear how a Tangible User In-
611 terface (TUI) is an easy and effective way of interacting with the system
612 throughout the composition of a workflow. Even though all our participants
613 are Computer Science undergraduates, their second-year group project is
614 their first chance of tackling a wider problem solving scenario, unlike their
615 first year’s individual development of smaller applications. This more com-
616 plex project required them to learn abstraction and decomposition skills,
617 whilst collaborating with peers. Using the puzzle metaphor and workflows

618 together with tangible interaction seemed to help them build the required
619 Computation Thinking skills: for instance, collaboratively planning and de-
620 signing the application’s tasks and assigning them to each participant seems
621 like a suitable scenario to practice abstraction and composition skills. More-
622 over, as with API development, the recipe metaphor provides different levels
623 of transparency and abstractions useful to generalize the problem whilst as-
624 sembling a puzzle might help with decomposing a bigger problem into smaller
625 ones [59].

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

636 From the interaction point of view we noticed one interesting remark made
637 by one of the participants: continuously tracking the smartphone’s position
638 on the surface using a fiducial marker requires the user to not cover its display
639 with her hand when moving it; however, the user’s hand position on the
640 smartphone might depend on her posture: if the user is standing straight, it
641 feels more “natural” to hold it from above — thus covering the fiducial marker
642 with her palm — while if sitting down, the user might feel more comfortable

643 grabbing it from the side, without covering its display, allowing its movements
644 to be tracked. Because the majority of existing smartphones are shaped in
645 the same way, it is worth studying this effect in more detail, in order to
646 establish whether we could provide users with a physical enclosure affording
647 the “right” way of holding the smartphone or whether it is a negligible effect
648 when the system runs on horizontal displays of a certain distance from the
649 floor.

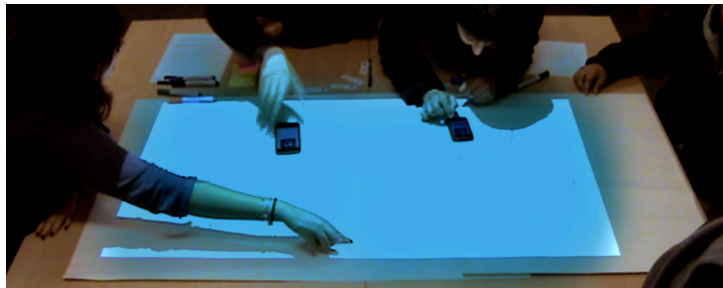


Figure 6: One of the participating groups to our study working with TAPAS.

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

657 *4.2. Designer Study*

658 We also interviewed three interaction design experts to get feedback on
659 the modality we have implemented in TAPAS; we carried out the interviews

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

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

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



Figure 7: The designer study setting.

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

687 Some of their suggestions focused on the way TAPAS presents data to
688 users and the use of the dynamic widget to get some input from them: due to
689 the kind of data handled right now — namely lists of files within directories
690 or book titles in a database — it makes sense to prompt users to choose an
691 option from a list or offer a keyboard to input raw text. Nevertheless, this
692 will not be the case if we have to deal with more structured data types, such
693 as points of interest on a map: therefore, they suggested that due to the com-
694 plexity of workflows that might be put together by final users, widgets might
695 be designed to be more flexible and personalizable depending on the two-fold
696 level of interaction between the user perspective and data perspective related
697 to the specific data handled by the widget. They emphasized that the two
698 perspectives are interlinked and reinforced mutually. We propose to consider
699 elements of human-centered information visualization in the redesign of the

700 widgets for the next interaction prototype; for instance, by following visual
701 metaphors that incorporate semantic relationships of visual objects both in
702 the physical (tangible) and virtual (digital) world [60, 61].

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

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

723 Indeed, there are clearly positive elements in our design for End-User
724 Development (EUD) of workflows to adapt public display services to users’

725 needs, such as the puzzle metaphor, the use of the smartphone as being tan-
726 gible and personal, and the ease of prototype deployment in-the-wild due
727 to its low-cost and flexible architecture. Nonetheless, there are some major
728 challenges to be addressed in future in terms of interaction design require-
729 ments, such as the flexibility/programmability of widgets and improving the
730 puzzle metaphor to highlight available functionalities.

731 **5. Discussion**

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

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

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

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

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

758 This contrast is also evident from the literature (see section 2.2) highlight-
759 ing the many differences between the Programming by Demonstration (PbD)
760 and Programming by Instruction (PbI) paradigms: due to its very nature,
761 when a system exploits PbD, the composition and execution environments
762 are perfectly overlapped, i.e. the same artifacts the users operate on to pro-
763 gram the system are used also to interact with its results, as with Robot
764 Programming by Demonstration; in Robot Programming by Demonstration
765 users teach movements to a robot by simply simulating them directly onto
766 its body. This is radically different from a PbI approach, where the two
767 environments — composition and execution — are generally detached from
768 one another, each one using different metaphors and concepts, e.g., in Yahoo!
769 Pipes there is a visual editor for composing a pipe (data-flow) that generates
770 a specific execution environment made of Graphical User Interface (GUI) el-
771 ements as designed by the user. While this distinction might be overlooked
772 from an interaction perspective when a system only relies on a GUI, it be-
773 comes more relevant when it is about Tangible User Interfaces (TUIs). Even

774 though PbI seemed the right paradigm to choose in our scenario due to its
775 generalizability and the benefits brought to Computational Thinking skills,
776 we argue that choosing the right paradigm according to the naturalness of
777 interaction is clearly scenario-dependent, as is often the case with Domain
778 Specific Visual Languages.

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

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

797 A final remark concerns the problem we were investigating first, namely
798 fostering the long-term appropriation of Pervasive Displays by enabling users

799 to repurpose them through EUD: during our first study we collected and
800 clustered the requirements of a typical scenario where Pervasive Displays
801 could already be used, but — due to their maintenance issues and progressive
802 loss of interest by users — are not yet widespread. Our analysis reported
803 three types of activities that end users need to be able to carry easily out with
804 a Pervasive Display in order to properly support user needs in the scenario
805 we considered: (1) scheduling, (2) reporting, (3) and discussion activities.

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

823 Finally, as we previously stated, it is undoubtedly worth pointing out the

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

833 **6. Conclusion**

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

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

848 We evaluated TAPAS by carrying out a two-phase study, the first phase
849 involving end users in a specific scenario — second year undergraduates work-
850 ing in groups — and the second phase with interaction designers. From the
851 first study’s results, it seems that our prototype provides a positive user ex-
852 perience and could be used in a collaborative project scenario where people
853 work together to tackle a complex problem; a potential side effect caused by
854 employing our prototype might be a development of Computational Think-
855 ing skills, thanks to our design rationale. However, from our findings it
856 also appears that coupling tangible interaction with a Programming by In-
857 struction paradigm causes an incompatibility of interaction styles between
858 the composition and the execution environments, where the use of a differ-
859 ent tangible-based syntactic construct in the former causes the need for a
860 different interaction style to be used in the latter.

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

872 In the future we plan to study in more detail issues arising from our find-

873 ings, with particular attention to the main challenges discussed in section 5.
874 We plan to exploit the feedback obtained from our studies in the next iter-
875 ation of TAPAS’ design and carry out additional evaluation studies in other
876 public scenarios, such as university settings or urban areas, and in non-public
877 collaborative contexts too, e.g., within a company. Moreover, further studies
878 will be carried out in order to draw more definitive conclusions regarding the
879 effect of the proposed interaction modality on the development of Compu-
880 tational Thinking skills, as well as within a fully in-the-wild setting, where
881 participants will be prompted to use the system without any researchers’
882 intervention. We also plan on studying whether extending TAPAS’ function-
883 alities without support for more complex workflows, as suggested by designers
884 and users, might improve its adoption.

885 **Acknowledgment**

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

891 **References**

- [1] H. Ishii, B. Ullmer, Tangible bits, in: the SIGCHI Conference, ACM Press, New York, New York, USA, 1997, pp. 234–241.
- [2] T. S. Mcnerney, From turtles to Tangible Programming Bricks: Explo-

- rations in physical language design, *Personal and Ubiquitous Computing* 8 (2004) 326–337.
- [3] 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.
- [4] D. Wang, T. Wang, Z. Liu, A tangible programming tool for children to cultivate computational thinking, *The Scientific World Journal* 2014 (2014) 1–10.
- [5] D. Wigdor, D. Wixon, *The Spatial NUI*, in: *Brave NUI World*, Elsevier, 2011, pp. 33–36.
- [6] 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.
- [7] A. Bellucci, A. Malizia, I. Aedo, Light on horizontal interactive surfaces: Input space for tabletop computing, *ACM Computing Surveys* 46 (2014) 1–42.
- [8] C. Ardito, P. Buono, M. F. Costabile, G. Desolda, Interaction with large displays: A survey, *ACM Computing Surveys* 47 (2015) 1–38.
- [9] 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).

- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [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. Burlison, 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. Evéquo, 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. Kubitzka, 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. Kubitzka, 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.