User-driven Design Guidelines for the Authoring of Cross-Device and Internet of Things Applications

Audrey Sanctorum
asanctor@vub.be
Web & Information Systems Engineering Lab
Vrije Universiteit Brussel
Brussels, Belgium

Suzanne Kieffer
suzanne.kieffer@uclouvain.be
Institute for Language and Communication
Université catholique de Louvain
Louvain-la-Neuve, Belgium

Beat Signer
bsigner@vub.be
Web & Information Systems Engineering Lab
Vrije Universiteit Brussel
Brussels, Belgium

ABSTRACT

Recently we see an increasing number of solutions for the authoring of cross-device and Internet of Things (IoT) applications. While most of these authoring tools have been realised based on existing metaphors and evaluated in subsequent user studies, there is no consensus on how to best enable end users to manage and interact with their devices, IoT objects and services. In order to establish some common guidelines for the development of cross-device and IoT authoring tools, we conducted an elicitation study exploring a user's mental model when defining cross-device interactions in IoT environments. Existing authoring solutions have further been checked for compliance with our guidelines and we developed a fully-compliant end-user authoring tool for cross-device and IoT applications. The presented guidelines may inform the design of new as well as the improvement of existing solutions and form a foundation for discussion, future studies and refinements within the HCI community.

CCS CONCEPTS

• **Human-centered computing → User studies**; **Interaction design**; User interface programming.

KEYWORDS

End-user development, cross-device interaction, Internet of Things, elicitation study, metaphors

ACM Reference Format:
Audrey Sanctorum, Suzanne Kieffer, and Beat Signer. 2020. User-driven Design Guidelines for the Authoring of Cross-Device and Internet of Things Applications. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society (NordiCHI '20), October 25–29, 2020, Tallinn, Estonia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3419249.3420136

1 INTRODUCTION

The design and development of user interfaces has always been difficult and time-consuming for both designers and developers [28]. Since the early days of graphical user interfaces (GUIs), various metaphors have been used in design to lower the complexity of a user interface (UI) by building on concepts that users are already familiar with from other domains [4, 24]. With the rapidly growing number of devices and smart things, new metaphors have been designed together with a multitude of dedicated applications to control these devices and smart objects. While research in the field of cross-device interaction (XDI) aims to facilitate the interaction between different devices through new interaction techniques, research in the domain of the Internet of Things (IoT) focuses on improving the interaction and communication between smart objects. However, the development of applications coping with interactions across devices and the wide variability of user needs remains a challenge. Therefore, more attention should be paid to make systems *easy to develop* rather than only *easy to use* [26]. There is a need for meaningful abstractions and metaphors to hide the low-level details and allow end users to become part of the development process [31]. Any programming efforts for new cross-device and IoT applications should be minimised by turning the development into an authoring rather than a programming activity [37].

The main goal of end-user development (EUD) is to enable non-technical users to create, modify and extend software artefacts through a set of methods, techniques and tools [26]. While many XDI and IoT authoring tools have been realised based on existing metaphors and subsequently been evaluated to investigate their usability [2, 6, 32], only a few prototype designs have been informed by end-user studies. Dey et al. [13] and Ghiani et al. [16] interviewed some end users to inform the design of their prototypes. Desolda et al. [11] performed an elicitation study with computer scientists to identify adequate visual composition mechanisms for the creation of rules, while others analysed related work to define guidelines for their prototype [35]. Further, as part of their XDI research, Nebeling [29] performed multiple studies to observe how end users interact with multiple devices and adapt the UIs depending on various device combinations. Other studies have been conducted in XDI research, involving informative, technical and heuristic evaluations [3]. However, to the best of our knowledge no elicitation study to learn more about the mental models of end users in combined cross-device and IoT settings has been carried out so far. We therefore conducted an elicitation study to close this knowledge gap. Since both XDI and IoT interaction form an integral part of our daily life, it is desirable to support both activities via a single unified solution rather than having a fragmentation of controls over different XDI and IoT solutions [38].

NordiCHI '20, October 25–29, 2020, Tallinn, Estonia

ACM ISBN 978-1-4503-7579-5/20/10…$15.00
https://doi.org/10.1145/3419249.3420136

We first present the methodology and the results of an exploratory study aimed to elicit the mental models that users apply when defining cross-device interactions in IoT environments. Based on the results of our elicitation study, we define a set of design guidelines for cross-device and IoT end-user authoring tools. We further discuss the proposed set of guidelines by checking to what extent related work follows the proposed guidelines. Finally, we present our work on an end-user authoring tool for XDI and IoT applications that is fully compliant with the presented design guidelines.

2 METHODOLOGY

2.1 Method

We conducted an exploratory elicitation study since our objective was to gain insight into the abstractions or *metaphors* that people use or rely on when thinking about XDI and IoT applications. In order to trigger participants' intent of design and engage them in design activities [13], we provided them with a scenario including both XDI and IoT concepts, and asked them to represent the interactions described in the scenario graphically on a sheet of paper. We chose this scenario-based approach rather than user tests with a prototype, as scenarios are evocative, promote reflection, analysis and innovative thinking [34]. In order to maintain an unbiased position as researchers, we followed the grounded theory method [5] for the data collection and analysis.

2.2 Elicitation Scenario and Questions

The scenario involved a character named Alex, a student and fitness enthusiast, who interacts with various cross-device applications and IoT devices. We used PowerPoint to present the elicitation scenario (see Figure 1) and trigger experimental tasks via a set of questions displayed on the last slide[1]. An example question was: "*How would you graphically represent the functionality and interactions between the different components in this scenario? For example, how would you draw that the interaction with Alex' phone triggers the TV to turn on?*"

A pilot study involving four participants, whose answers were not used in the final study, allowed us to review and improve both the elicitation scenario and the questions. We made the study material available to the participants on paper (two slides per page) in French or in English, allowing the participants to perform the experimental tasks either in French or in English.

2.3 Participants

We recruited 30 participants (12 females and 18 males) with an average age of 33.8 years (min=23, max=76, SD=11.4). Half of the participants (15) had a technical background, meaning that they had or were studying engineering sciences. The other half of the participants had a non-technical background (Table 1). This mixed sampling was necessary since we wanted to verify which metaphors would fit the mental models of both technical as well as non-technical users. Most participants (27) had already heard about the term IoT and 15 participants had some IoT devices at home. Only slightly more than half of the participants (17) knew the term cross-device interaction. One participant did not possess any smart device, 10 participants had only a smartphone and 19 had two or more smart devices.

[1]All study material is available at https://gitlab.wise.vub.ac.be/asanctor/iot-xdui-elicitation-study

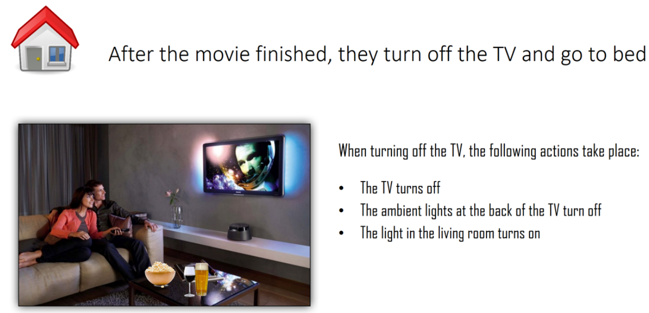


Figure 1: Example slide from the Scenario.pptx presentation

Table 1: Background of participants

Background	Participant
Technical	P1, P2, P8, P9, P11, P13, P16, P21, P22, P25, P26, P27, P28, P29, P30
Non-technical	P3, P4, P5, P6, P7, P10, P12, P14, P15, P17, P18, P19, P20, P23, P24

2.4 Experimental Protocol

Two experimenters carried out the elicitation study in 15 sessions, each involving two participants; five sessions where both participants had a technical background, five sessions where both participants had a non-technical background, and five sessions where one participant had a technical background and the other had a non-technical background. First, both participants had to fill in a consent form and received a short presentation about the concepts of XDI and IoT. Then, they received instructions to perform the experimental tasks and have been asked to read the elicitation scenario and answer the elicitation questions by drawing their answer while speaking their thoughts aloud, in accordance with the *think-aloud* protocol. During the execution of the experimental tasks, each participant sat in a separate room with one experimenter each. Once participants finished their drawings, experimenters asked them to compare their solutions, make possible improvements, and present a unified solution. This was followed by a short interview with one participant at a time. Finally, one experimenter asked the participants to fill in a questionnaire. We videotaped each session for later use during the analysis and the experimenters were taking notes during the entire study.

2.4.1 XDI and IoT Presentation (10 min). One experimenter consistently gave the short presentation introducing the concepts of XDI and IoT, with examples of both XDI (Spotify and Chromecast) and IoT (Philips Hue and Amazon echo). In order to avoid bias in the experiment, participants were not provided any examples of metaphors.

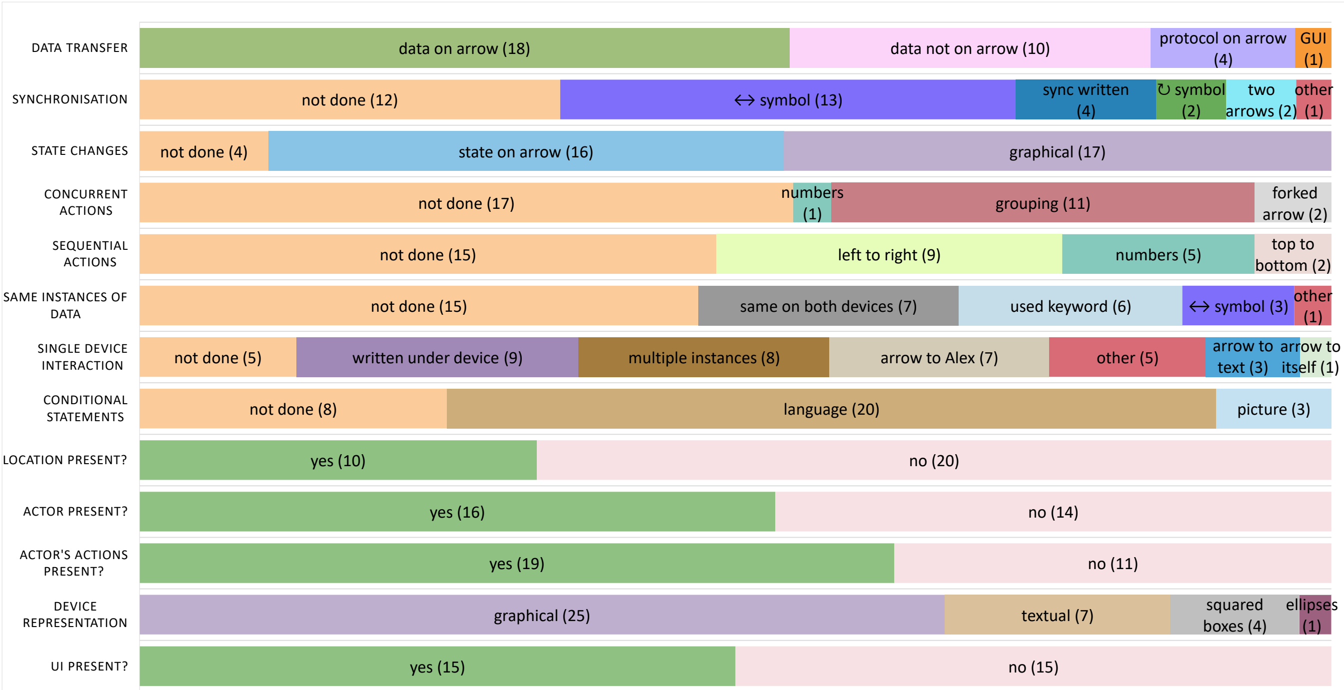


Figure 2: Summary of results

2.4.2 Scenario Drawing (30–60 min). Each participant received a hardcopy of the elicitation scenario and questions, blank paper sheets, pens and pencils. At any point, participants could ask questions to the experimenter. In case of comprehension problems of the elicitation questions, experimenters would reformulate the question according to their script. While observing the participants, experimenters could also ask questions to clarify some of their drawings such as *"Are these actions happening at the same time? How did you indicate that?"*. These questions evolved over the course of the study as suggested in [5] and based on our analysis after each session, more directed questions such as *"How did you indicate that the movie playing on Alex' phone is the same one as the movie shown on the TV?"* could be asked.

2.4.3 Comparison of Drawings (30 min). Once participants finished their drawings, experimenters asked them to compare their solutions, make possible improvements, and present a unified solution. This way, we could get more insights on why some participants chose to draw certain concepts in a certain way and see whether, by comparing the drawings, they could improve their own drawing or make a new one combining elements from both participants. Again, at any point participants could ask questions to the experimenters. If the participants had difficulties to start the comparison, the experimenters would ask a few questions according to their script.

2.4.4 Interview and Questionnaire (10 min). One experimenter interviewed each participants separately and asked five questions. Questions Q1 to Q3 were related to participants' drawings and their potential difficulties during the study. Question 4 investigated whether participants thought they had enough control over existing cross-device and IoT solutions. Question 5 investigated whether they had further comments about the study. Participants also filled in a questionnaire consisting of 15 questions. Five questions were dealing with demographic data and participants' education. Nine questions aimed to collect information about their exposure to XDI, IoT and technology in general, while a final question asked them for feedback about the ease of completing the study.

2.5 Data Collection and Analysis

The collected data includes the notes taken by experimenters, video recordings, participants' drawings, interview transcripts and the answers filled in by the participants in the post-study questionnaire. All participants completed the user study. The analysis of their drawings allowed the two experimenters to document a coding guide for the drawing components. An unbiased position was guaranteed by following the grounded theory method [5] for the data collection and analysis. We started by comparing the first batch of drawings based on the drawing itself but also on think-aloud, memos and interview data (questions Q1 to Q3) that we had collected. From this comparison we identified different characteristics (tagged as codes), based on which different concepts arose. By comparing those concepts, we grouped them into categories. For example, we had the "use of frames" or "use of '+' between actions" concepts, which we grouped into the "use of grouping" subcategory that—depending on their properties—became part of the "concurrent actions" or "sequential actions" category. We continued this comparison process until no new concepts or categories could be found. Our findings are summarised in Figure 2. Each colour represents a specific subcategory; some subcategories might overlap and therefore not sum up to the total number of 30 participants for each row.

While we primarily conducted a qualitative study, we still checked for any correlation and differences per category between participants with or without a technical background. Unfortunately, there is not enough data in every category/subcategory to run statistically significant tests. However, there is a fairly high correlation between both groups of participants except for sequential actions which seem to be mainly used by participants with a technical background. The Pearson correlation coefficients varied between 0.65 and 0.95, with 0.12 as lowest value for sequential actions. Finally, on a side note, while the *comparison of drawings* phase allowed participants to improve their drawings, only minor changes where made during this part of the study.

3 RESULTS

In the following we describe each category presented in Figure 2 in the order of its appearance. We use the format $p(t, n)$ to provide information about participants, with p representing the total number of participants which can be divided in t participants with a technical background and n participants with a non-technical background. Note that while the first interview questions helped for the drawing analysis, the answers to the remaining questions served as an indication on whether participants would be interested in an authoring tool and whether they had enough customisation support in their current applications. The results to these questions were quite mixed with 16 participants interested in a tool and 16 participants expressing a lack of control over their applications [36].

3.1 Data Transfer and Synchronisation

All but one participants (29 (14,15)) used arrows at least once to indicate data transfer and interaction between devices and smart objects. Amongst these 29 participants, 20 (11,9) annotated arrows with either data to be transferred (18 (10,8)) or to highlight the transfer protocol directly on the arrow (4 (3,1)), while 10 (3,7) did not highlight such transfer on any arrows, but rather on the devices (3 (0,3)) or next to the devices (2 (1,1)). Further, 4 (2,2) drew an arrow from the data to the device in order to define the data as input into the device (Figure 3). Surprisingly, 1 (1,0) participant only added arrows during the comparison of drawings to clarify how certain events trigger specific actions. This participant further mentioned having difficulties to show that devices were connected and used an ad-hoc synchronisation symbol on multiple devices.

Participants used different kinds of arrows, although not in a consistent way. For instance, participant P12 used dashed and solid arrows and mentioned that they were both used for the same purpose. Some participants used radio waves in combination with arrows as shown in Figure 4e. A few participants also using double-lined arrows (⇒). Slightly less than half of the participants (12 (8,4)) did not make any clear distinction between transferring and synchronising data between two devices. More concretely, 13 (4,9) used a double-sided arrow (↔) to indicate synchronisation between two

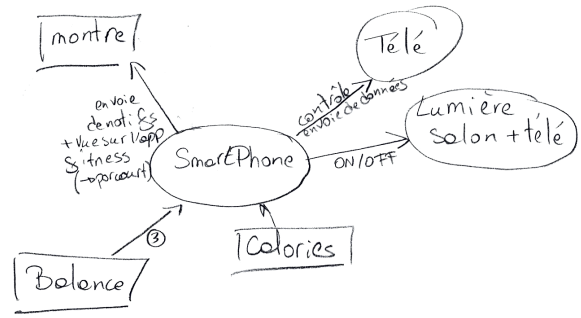


Figure 3: Participant P15 using an arrow to define *'calories'* as input data into the device

devices, while only 2 (1,1) used a synchronisation symbol (↻). Further, 4 (3,1) participants used the *"sync"* keyword and 2 (1,1) used two opposite arrows as illustrated in Figure 4a.

3.2 State Changes

Many devices change state (e.g. switching between an 'on' and 'off' state) according to events and actions of the elicitation scenario. In order to go from one state to another, 16 (9,7) participants used regular arrows labelled with the corresponding command for changing the state (e.g. *"turn on"*). Moreover, 17 (9,8) participants represented the state changes graphically, for example by showing a light bulb emitting some rays of light. Only 4 (0,4) did not clearly show any state changes.

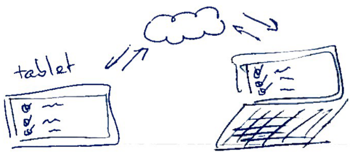
3.3 Time-based Actions

Not all participants made a distinction between the time-based actions (i.e. concurrent versus sequential actions) involved in the elicitation scenario. Concurrent actions were grouped together by 11 (5,6) participants, either by noting the concurrent actions on the same arrow or by grouping the actions in one drawing. Figure 3 depicts grouping using *"+"* both on an arrow and in an ellipse: *"lumière salon + télé"*. 2 (2,0) participants used forked arrows as shown in Figure 4c and only 1 (1,0) participant used the same numbers next to actions to indicate that they happen at the same time.

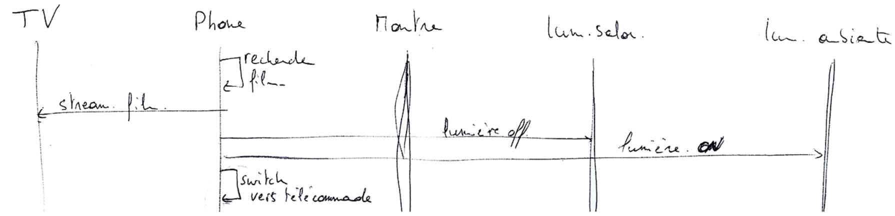
Half of the participants (15 (5,10)) did not clearly indicate sequential actions. In contrast, 9 (8,1) participants noted down the actions from left to right with an arrow in between, 1 (1,0) did not use arrows in between but indicated that time was going from left to right at the top of the drawing. Further, 2 (1,1) participants indicated time going from top to bottom, one of them using a sequence diagram as illustrated in Figure 4b. Finally, 5 (2,3) participants used numbers to indicate the order of actions.

3.4 Multiple Instances of the Same Data

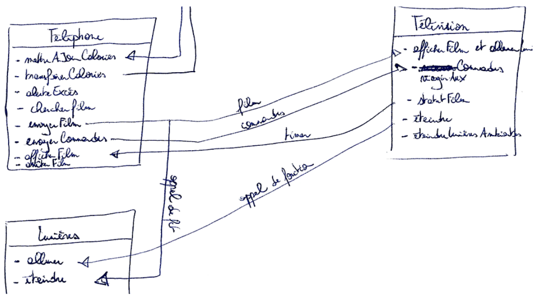
The elicitation scenario involved a movie shown on both the TV and Alex' smartphone. These two instances of the movie are supposed to be synchronised. Half of the participants (15 (8,7)) did not clearly indicate multiple instances of the same data. In contrast, 3 (0,3) participants added a double-sided arrow, 6 (3,3) used keywords such as "copy", "cast split", "duplicate" or "||" to show that the movie



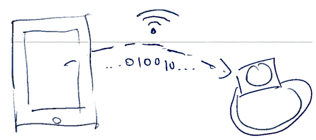
(a) Synchronised data (P25)



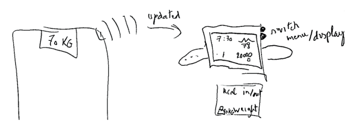
(b) Time-based actions in sequence diagram (P11)



(c) Forked arrows (P9)



(d) Use of Wi-Fi symbol (P28)



(e) Data transfer (P21)

Figure 4: Interaction drawings of different participants

was duplicated on the phone, 7 (3,4) drew or wrote the same thing on or next to both devices as shown in Figure 4a with the same drawing on the laptop and tablet. Finally, 1 (1,0) participant just wrote down the option that the film can be shown on both the TV and smartphone.

3.5 Single Device Interactions

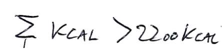
The elicitation scenario involved interactions occurring on one device only: e.g. the phone becoming a remote controller and then a movie viewer, or the smartwatch monitoring Alex while running. To illustrate such actions, 8 (6,2) participants drew an arrow from one instance of a device to another, 7 (2,5) participants drew an arrow from the device to Alex, 3 (1,2) drew an arrow going from the device to some text describing what was happening on this device and 1 (1,0) drew an arrow originating and ending at the same device, as shown in Figure 4b for some of the phone's interactions. In contrast, 9 (4,5) participants wrote down the interactions below the device.

3.6 Conditional Statements

The elicitation scenario involved conditional statements such as *"Since Alex exceeded the 2200 kcal today, she receives a notification on her smartwatch warning her about this excess"*. In order to highlight simple conditions, 15 (7,8) participants wrote data ">2200 kcal" on or in between arrows (Figure 5a) while 5 (3,2) used keywords such as "IF" ("SI" in French) and "WHEN" (Figure 5c). A last group (3 (3,0)) drew the conditions (Figure 5b).

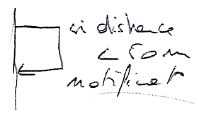
3.7 Location

In the elicitation scenario the location of the interaction did not play a role. However, 7 (1,6) participants still wrote down the location names where the interaction took place. Amongst them, only 1 (0,1) mentioned that the location mattered for the interactions. Instead of writing down the location, 3 (1,2) participants mentioned some form of location awareness by, for example, drawing a sensor.



(a) Simple data (P12)

(b) Drawing (P26)

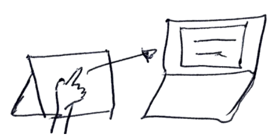


(c) Keywords (P11)

Figure 5: Examples of conditions

3.8 Presence of Actors

The drawings of 16 (6,10) participants included the presence of Alex, either drawn or in textual form. Further, 2 (2,0) participants only drew Alex' hand as shown in Figure 6.



(a) Participant P13 showing Alex using a swipe

(b) Participant P20 showing Alex using drag-and-drop

Figure 6: Example of Alex interacting with her tablet

3.9 Actor's Interactions

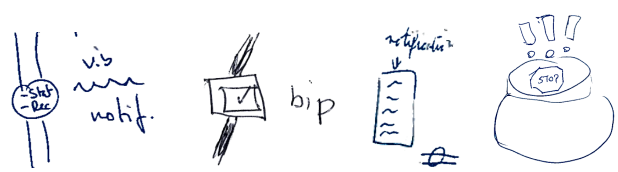
Many participants highlighted interactions such as "swipe", "touch" or "press" between Alex and her devices: 19 (9,10) by either writing the type of interaction next to Alex, on an arrow or by drawing Alex performing the action (Figure 6b). 6 (3,3) of them only represented the interaction graphically (Figure 6a). Another 2 (0,2) just drew an arrow from Alex to a device without specifying the type of interaction.

3.10 Representation of Devices

Participants represented devices and smart appliances either graphically or in textual form. 25 (12,13) participants used a realistic graphical representation of the devices (Figure 4a, Figure 4d and Figure 4e), 4 (2,2) wrote the device names into rectangular shapes, 7 (3,4) only used the name of the devices, and 1 (0,1) wrote the device names into ellipses. Some participants were not consistent in the representation of devices, switching between graphical and textual form. Participant P15 even mixed the use of squares and ellipses to represent devices (Figure 3). When devices were represented graphically, 15 (8,7) participants drew certain UI elements such as buttons for triggering actions on the devices. This is surprising given that the scenario never mentioned any buttons and we asked the participants to focus on the interaction between devices rather than the UI of an individual device.

3.11 Use of Symbols and Keywords

It is interesting to analyse the symbols and keywords used by participants, since only 6 (3,3) did not use symbols other than the devices and arrows. The Wi-Fi and Bluetooth symbols were sometimes used to indicate whether the connection was made via Wi-Fi or via Bluetooth (Figure 4d). The moon and sun were used by 2 (0,2) participants to indicate day and night. Notifications were often shown by drawing a vibration symbol next to or on a device with the corresponding message usually shown on the device but also symbolised in different ways (Figure 7).



(a) Vibration (P25) (b) Beeping (P13) (c) Text (P26) (d) Stop (P28)

Figure 7: Notification examples

4 DESIGN GUIDELINES

Based on our findings, we formulated eight guidelines (G1 to G8) for cross-device and IoT end-user authoring tools and systematically checked existing solutions for compliance with our guidelines as highlighted in Table 2. Our initial guidelines might be extended and further refined over time. The guidelines should help developers to create end-user authoring tools based on what end users prefer, understand and expect when dealing with XDI and IoT interactions.

G1: Use pipeline or graph metaphor to represent interactions in an end-user authoring tool, as participants widely used arrows to represent interactions between devices and smart things (29 participants). The pipeline metaphor graphically represents applications as directed graphs where nodes correspond to elementary services with interconnecting links (i.e. pipelines) [9]. To prevent end users from linking incompatible devices and services while using the pipeline metaphor, a specific colour should highlight compatible inputs and outputs, as done in [8]. Finally, pop-up windows can be used to gather data about the configuration of an interaction [11].

Although G1 seems obvious, almost none of the authoring tools found in the related work actually follows this guideline as illustrated in Table 2. This might be due to the fact that this metaphor requires more cognitive effort [7, 11]. However, as shown by our study results, it seems to be the first thing popping into people's mind when speaking about interaction across devices.

G2: Assign different arrow types to different types of interaction, including uni-directional arrows to represent regular interactions (29 participants) and double-sided arrows to represent synchronisation specifically (13 participants). Optionally, a difference can be made between user-initiated triggers (e.g. button press) and contextual triggers (e.g. time) in order to easily identify the type of interaction present in the authoring environment. One could use dashed arrows for contextual triggers, given that our participants used them second most.

To the best of our knowledge, none of the existing authoring tools supports guideline G2. This might be due to the fact that it is closely related to G1. While the E-Wired prototype supports the pipeline metaphor, it does not provide different arrows for different kinds of interaction [11].

G3: Provide a realistic graphical device representation of the available devices, as most participants represented devices and smart objects by realistic graphical representations in their drawings (25 participants). The participants who did not, mentioned that they would have done so if they had better drawing skills. Ideally, a user should be able to choose the graphical representations in order to easily recognise and distinguish devices. Further, this representation offers the possibility to show some graphical UI components on the device itself.

Guideline G3 is the most supported guideline in related work. The use of colour codings as explained in G1 in combination with the visual representation of devices (G3) might further reduce the cognitive effort.

G4: Provide a graphical representation of users either as individuals or groups of users, given that 16 participants depicted Alex in their drawings. Optionally, the action performed to trigger an interaction, such as pressing a button, should be expressed, as 19 participants drew Alex' actions (e.g. swiping). This representation of users should also be used to show contextual interactions involving the user, such as *"If Alex is at home, turn on the Wi-Fi on her phone"*.

As shown in Table 2, only three of the existing authoring tools fully support G4. In contrast, devices often have a realistic graphical representation.

G5: Represent sequential interactions from left to right and group concurrent interactions, given that 9 out of the 15 participants who represented time in their drawings used arrows from

Table 2: Compliance of existing XDI and IoT authoring tools with guidelines G1 to G8. Authoring tools are listed in alphabetical order for both XDI (top) and IoT (bottom).

Systems	G1	G2	G3	G4	G5	G6	G7	G8
ACCORD [34]	○	◌	●	●	◖	◖	◌	◖
Direwolf [21]	○	◌	○	○	◌	◌	◖	◖
Ghiani et al. [15]	○	◌	●	●	◖	◖	?	◖
Interplay [28]	○	◌	○	○	◌	◌	◌	○
Jelly [27]	○	◌	○	◌	◌	◌	●	◖
Platform Composition [33]	◖	◌	◖	○	◌	◌	◌	◖
SmartComposition [22]	○	◌	○	◌	◌	◌	◖	◖
XDBrowser 2.0.0 [30]	○	◌	●	◌	◌	◌	◖	○
a CAPpella [12]	○	◌	●	○	●	◌	◌	○
AppsGate [6]	○	◌	○	○	◖	◖	◌	○
Atooma[2]	○	◌	●	○	◖	◖	◌	◖
Direwolf 3.0.0 [20]	○	◌	?	?	?	?	?	?
E-Wired [11]	●	○	○	?	●	◖	◌	○
EPIDOSITE [25]	○	◌	○	○	○	○	◌	○
HomeRules [36]	○	◌	●	?	●	◖	◌	◖
iCAP [13]	○	◌	●	●	◖	◖	◌	◖
IFTTT	○	◌	●	○	○	◖	◖	◖
ImAtHome [14]	○	◌	●	◖	○	◖	◌	◖
Keep Doing It [10]	○	◌	●	○	◖	●	◌	◖
Puzzle [8]	○	◌	○	?	●	◖	◌	○
SmartFit [2]	○	◌	○	○	◖	○	◌	○
TARE [16]	○	◌	○	○	○	○	?	○
Tasker[3]	○	◌	●	○	◖	◖	●	◖
TouchCompozr [23]	○	◌	?	?	○	◖	◌	○
Versatile [17]	◖	◌	○	◌	○	○	◌	○
Visit [1]	○	◌	○	◌	●	◖	◌	◖
Zipato[4]	○	◌	●	?	◖	◖	◌	◖

Legend
○ Not fulfilling guideline
◖ Partially fulfilling guideline
● Completely fulfilling guideline
◌ Functionality not present
? Not specified

left to right. For more complex interactions, optional numbering could be used to avoid confusion (5 out of these 15 participants). We also recommend grouping to represent concurrent actions and triggers, either by grouping the actions on one arrow if possible or by representing the action components below each other (optionally framed). Grouping was used by 11 participants and is further in line with the *similarity* and *proximity* Gestalt principles [19].

Twenty of the authoring tools identified in related work support the notion of time with 70% of them partially or fully satisfying G5. The authoring tools which do not fulfil G5 (30%) often rely on a textual (non-graphical) form to represent time from left to right or from top to bottom.

G6: Provide textual as well as graphical representations for conditional statements, as 20 participants mixed both representations in their drawings by, for example, writing down conditional statements on arrows, in between arrows or on devices. This guideline is compliant with existing literature which advocates for both visual and textual representations for a better understanding of conditional statements [13]. End users should be able to freely switch between those two representation forms. If-Then statements could be used as textual representations, as inexperienced users can quickly learn If This Then That (IFTTT)[5] to create programs containing multiple triggers or actions [39]. Desolda et al. [11] provide further guidelines for rule composition with the *Rule_5W* model. In addition, conditional statements should be represented via graphical elements supplemented by a textual condition (e.g. *"sum of calories > 2200 kcal"*).

Guideline G6 is never fully satisfied in related work, except for Keep Doing It [10] that provides both equivalent textual and graphical representations. Most authoring tools only partially satisfy G6; although they offer a mix of textual and graphical representations, they do not allow to switch between them. This is quite surprising since graphical representations lack specificity and do not include enough details about conditional statements.

[2]https://resonance-ai.com/about.html
[3]https://play.google.com/store/apps/details?id=net.dinglisch.android.taskerm&hl=en
[4]https://www.zipato.com
[5]https://ifttt.com

G7: Support UI design by offering users the possibility to create their own UI as in [15, 27], given that half of our study participants drew their own UI elements. This allows for more customisation and to create UI-triggered rules. Most authoring tools do not satisfy guideline G7 and if they do so, their support is limited to widgets only.

G8: Include (custom) symbols and support for annotations, given that 24 participants used icons and symbols in their drawings. The symbols and annotations should not be linked to a particular functionality, but could rather serve as a way to better understand and remember what is represented in the authoring environment.

G8 is at least partially fulfilled by most systems identified in the related work as highlighted in Table 2. However, none of the tools use symbols and annotations to simply add supplemental information to the authoring environment without affecting any interactions.

5 DISCUSSION AND RELATED WORK

In this section we discuss a selection of the more prominent related XDI and IoT end-user authoring tools, highlight the used metaphors and check for their compliance with our guidelines as summarised in Table 2.

5.1 Authoring of Cross-Device Applications

Over the years, many web-based tools have emerged, such as XDBrowser [29, 30], SmartComposition [22] as well as the work on DireWolf [21]. For example, XDBrowser enables end users to re-author web pages across devices by using a selection tool allowing them to select parts of a web page and copy or move the selection between devices. SmartComposition [22] and DireWolf [21] provide pre-built widgets that can be distributed across different devices via *drag-and-drop*. Jelly [27] offers more freedom in the interface design process by offering a design environment where UIs can be designed for multiple platforms in parallel by *copy and pasting* parts of a user interface from one device to another. While these tools all integrate the concepts of multiple devices, only XDBrowser shows these devices graphically by using icons and is therefore fully compliant with guideline G3. Since these tools do not include the notion of user, time and rules, guidelines G4 to G6 are represented by dashed circles in Table 2.

An authoring environment including context-dependent cross-device UIs has been presented by Ghiani et al. [15] allowing users to create contextual, adaptation and distribution *rules*. However, an evaluation revealed that end users without programming skills had difficulties in understanding the proposed concepts. Therefore, the authors proposed an extra layer on top of their tool which has been used to compare the tool to our guidelines. While rules are primarily represented textually in the main authoring environment, the extra layer provides a more graphical but less expressive view for creating rules. Since this graphical view does not offer a textual representation of rules, guideline G6 is only partially fulfilled. While the tool groups the triggers and actions in separate frames, the sequence of actions is not represented from left to right, making guideline G5 also partially fulfilled. It is unclear whether a simplified version of the UI creation part has been created and the conformance with guideline G7 is therefore unspecified.

Taking a step towards the authoring of IoT applications, Humble et. al. [18, 33] presented the ACCORD editor enabling end users to configure their ubiquitous computer environments based on the *jigsaw puzzle* metaphor. In contrast to the pipeline metaphor suggested in guideline G1, the jigsaw puzzle metaphor's expressiveness is limited by the number of sides of a puzzle piece. ACCORD provides a good visual overview of the interactions to the end users by showing devices and user actions, such as a "finger pressing a button". Guidelines G3 and G4 are thus fulfilled, time is also shown graphically from left to right by left-to-right couplings of puzzle pieces. However the grouping of concurrent interactions is not supported and guideline G5 is only partially fulfilled. While some textual explanation is provided below each puzzle piece in the menu, the connected pieces' configuration is only visible graphically, which makes it difficult to recall how the pieces are configured, and implies that guideline G6 is only partially fulfilled.

A variation of the jigsaw puzzle metaphor, is the *join-the-dots* metaphor used by the Platform Composition [32] technique, where the GUI shows devices as large circles enclosing smaller circles representing a device's core services. To create a connection between a service and a device, the user simply draws a line from the service to the device. The tool provides a graphical overview of all available devices and their services and outlines the entire system state. Nevertheless, since devices are represented as circles with their services represented by an icon, guideline G3 is not fully addressed. Further, users are only represented textually and guideline G4 is unfulfilled. Guidelines G5 to G7 cannot be evaluated since the notion of time, rule and UI creation is not present.

5.2 Authoring of Internet of Things Applications

Various commercial solutions allow users to configure their smart environments via *Event-Condition-Action (ECA)* rules. A well-known example is IFTTT, which enables users to create conditional statements that are automatically executed based on the internal state of apps or web services. IFTTT already supports certain IoT devices and can therefore also address emerging IoT devices as discussed by Ur et al. [39]. While IFTTT incorporates a graphical representation of their services and represents rules half textual and half graphically in the mobile interface, it does not offer switching between a textual and graphical representation and therefore does not fully comply to guideline G6. Similar cases are Atooma, HomeRules [35], ImAtHome [14] and Tasker.

Since time is not represented graphically from left to right, and there is no grouping of concurrent actions, IFTTT does not fulfil guideline G5. While IFTTT supports rules with only one event and one action, solutions such as Atooma, AppsGate [6], EPIDOSITE [25], HomeRules [35], iCAP [13], ImAtHome [14], Keep Doing It [10], Puzzle [8], SmartFit [2], TARE [16], Tasker, Visit [1] and Zipato can deal with more complex rules. However, only a few [1, 8, 35] depict the sequence of an interaction (trigger/action) as described in guideline G5, grouping concurrent actions and showing sequential actions graphically from left to right. Grouping has been done by using the '+' symbol in Atooma, Keep Doing It [10] and HomeRules [35]. iCAP [13] on the other hand uses frames to group concurrent triggers and actions.

Most IoT tools do not offer support for UI creation. IFTTT proposes a button widget that can be linked to some functionality and thus only partially fulfils guideline G7. Tasker goes a step further and supports the UI creation for pop-up screens on mobile devices. In the non-commercial tools, TARE [16] allows end users to perform actions on IoT appliances but in contrast to other systems, it also supports UI modifications and distribution. However as it is unclear whether the tool supports UI design, we marked G7 as not specified. Note that to the best of our knowledge TARE is the only authoring tool supporting both IoT interaction as well as cross-device interaction and UI distribution. Since the authoring environment of TARE is textual rather than graphical, it does not comply to our other guidelines. The same goes for EPIDOSITE [25] which uses programming-by-demonstration for automating mobile IoT applications. Kubitza and Schmidt [23] took a similar approach by proposing an IoT prototyping platform that simplifies the integration of devices and the control of smart environments. It combines traditional text-based programming with interactive physical programming-by-demonstration. Their mobile TouchCompozr GUI allows end users to form trigger/action rules by demonstration. For example, a physical switch can be defined as trigger by pressing a button on the mobile interface at the same time as the physical switch on the wall. Programming-by-demonstration has also been used in a CAPpella [12] which supports the creation of context-aware behaviour by demonstration with the start and end indicated via the *timeline* metaphor, thereby fulfilling guideline *G5*. A CAPpella also fulfils guideline G3 by showing a graphical representation of devices. However it does not fulfil guideline G4 since users can be recognised through RFID tags by the system but the UI does not reflect this information. Since a CAPpella is based on behaviour recognition and not on rules, the functionality for guideline G6 is not present. The main author of a CAPpella later proposed iCAP [13], a visual rule-based system to prototype context-aware applications for smart environments. While the visual UI is simple and intuitive for end users, it lacks some textual counterparts and does not satisfy guideline G6. Guidelines G3 and G4 are fulfilled while G5 is partially addressed as explained earlier.

AppsGate [6] helps end users to control and augment their home by creating *rules* via a pseudo-natural language. Further, a smarthome can be monitored via *timelines* and through a *dependency graph* showing the relations between entities. However, since the timelines and graph view cannot be modified and thereby have an impact on the created rules, guidelines G5 and G1 are not satisfied. In the rule authoring environment, grouping is supported but interactions are read from top to bottom making guideline G5 partially fulfilled.

IoT-MAP [17] is a smartphone solution that dynamically discovers devices, downloads the necessary software modules and provides the Versatile UI to the end users for the mashup and composition of smart things. The composition UI is based on Node-RED[6] and uses the *pipeline* metaphor. Since there are no arrows to indicate the direction of the data flow, guideline G1 is only partially fulfilled. On the other hand, E-Wired [11], one of Desolda et al.'s prototypes, fully fulfils guideline G1, but does not offer much graphical support. Interactions can be grouped and are represented from left to right in accordance with guideline G5.

Following a component-based web mashup approach, Koren and Klamma [20] presented an extension of the DireWolf framework [21] that integrates heterogeneous Web of Things (WoT) devices by including UI components directly served by WoT devices. Although the system sounds promising, little information is available on the use of the tool itself.

Finally, the *jigsaw puzzle* metaphor is used in Puzzle [8], which supports the development of IoT applications on smartphones. Thereby each puzzle piece represents some functionality that can be composed by connecting the pieces, with the shape and colour indicating the number of inputs and outputs as well as the data that can be exchanged. A commercial IoT solution making use of the jigsaw puzzle metaphor is the Zipato Rule Creator. Puzzle fully supports guideline G5 since it allows grouping and shows the interactions from left to right. Zipato, however, shows the interaction flow from top to bottom, but supports grouping, and therefore partially complies to guideline G5. Symbols are generally used by many tools but never to freely annotate the authoring environment, making guideline *G8* only partially fulfilled.

6 PROTOTYPE

Based on the presented design guidelines and the body of existing work, we developed a prototype of an end-user authoring tool for XDI and IoT applications[7]. The authoring tool is structured into four views, the *UI Design*, *Interaction*, *Rules* and *Home* view. The *UI Design* view allows users to design a GUI for their applications and has been inspired by Ghiani et al. [15] and Jelly [27]. Users can drag and drop UI elements on a device screen of their choice and customise their UIs as required by guideline G7.

In the *Interaction* view shown in Figure 8, users can define different kinds of interactions. The left sidebar contains elements that users can drag and drop to the authoring space on the right. The elements are grouped into five categories. The first one contains the *devices* consisting of smart devices and *things*. The second category contains *services* such as a *weather forecast* service. The next one regroups contextual elements, such as user, time and location. A fourth category contains the different types of arrows to define *interactions* between devices. Regular arrows for representing actions happening between devices, except for synchronisation which is represented via double-sided arrows in accordance with guideline G2. Dashed arrows are used for contextual interactions. The last category includes *symbols* and annotations. They do not offer extra functionality but, conforming to guideline G8, can help end users to better remember their defined interactions. Devices and users are represented graphically as required by our guidelines G3 and G4. One can switch between this and the previous view by pressing the magnifying glass on a device for getting details-on-demand. When selecting an arrow, elements that can be connected will be surrounded by green dots and the ones that cannot be connected with red dots in accordance with guideline G1. After selecting a source element by clicking on a green dot, they can follow the join-the-dots metaphor [9] and select another green dot surrounding the

[6]https://nodered.org
[7]https://youtu.be/NnQ9auXKj68

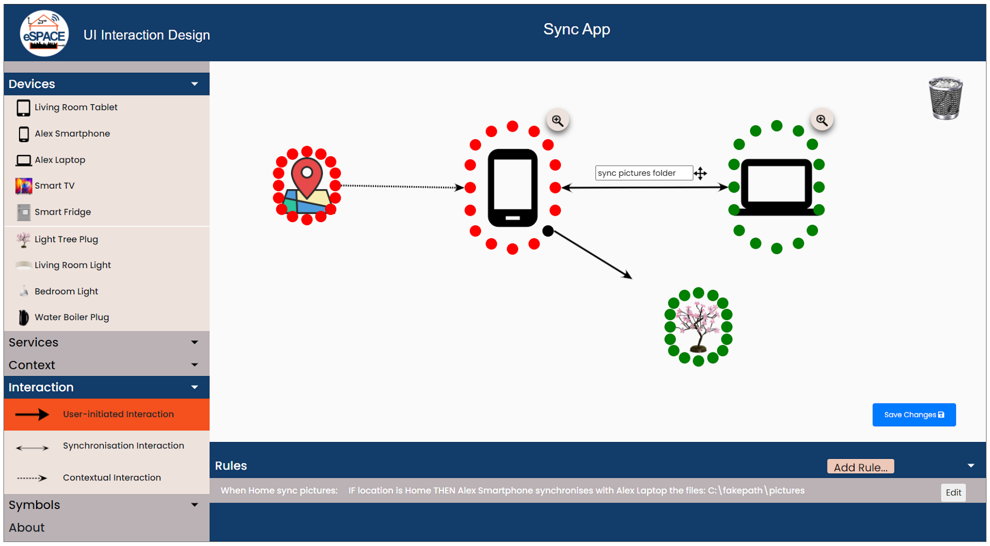


Figure 8: User creating an interaction in the Interaction view of the end-user XDI and IoT authoring tool

target device as shown in Figure 8. An arrow will then appear between both elements and depending on the type of arrow, a pop-up window (inspired by the work of Desolda et al. [11]) asks for parameters such as the data to be synchronised or the time when the synchronisation should take place. Exemplary interactions could be *"when I'm home synchronise my pictures between my phone and my computer"*, as illustrated in Figure 8. All interactions shown in the authoring space are also described in terms of trigger/action rules and shown in the bottom container of this view.

After clicking on a rule or the *add rule* button, users will be taken to the *Rules* view which can be used to create or modify a rule. Rules defined in this view are graphically represented in the *Interaction* view. The two views offer a consistent graphical and textual representation of rules as requested by guideline G6. Rules created in the *Rules* view will be shown graphically in the *Interaction* view following guideline G5. The *Rules* view has been inspired by Ghiani et al.'s [16] Trigger-Action Rule Editor that uses tiles to group elements belonging to the same category and also provides an overview of the current rule in a sentence above these tiles. We renamed some of the tiles according to the *Rule_5W* model [11]. The 5 "Ws" stand for *Which* services are involved in the rule, *Who* triggers the events and actions, *When* are they triggered and *Where*. The last "W" stands for *Why*, which is used to report a short description explaining the behaviour of the rule. Rather than using the 5Ws as such, we renamed them as follows in order to allow for a better understanding of the rule composition for end users: *Devices* (which), *Users* (who), *Time* (when), *Location* (where) and *Description* (why). Note that, *Devices* refers to services and devices as done by Wisner and Kalofonos [40] who believe that end users think more easily in terms of devices. Additionally, we added saved triggers and actions to promote re-use. Similar to the IFTTT recipes, we used *IF <trigger_expression> THEN <action_expression>* but also allow for complex expressions using the boolean AND, OR and NOT operators.

The *Home* view can be seen as a dashboard-like overview as seen in other authoring tools [6, 14, 16], enabling the regrouping of user-defined applications, rules and devices. This view has been created based on the comments of many participants who said that they would like to regroup applications rather than having to switch between them to use different smarthome appliances. Participants often also started by drawing an overview of the connected devices, but then continued with different more specific drawings.

Our initial prototype of an end-user authoring tool for cross-device and Internet of Things applications is fully compliant with the proposed design guidelines G1 to G8. While an initial study of our tool showed promising results [36], a more detailed evaluation will have to be conducted in order to fully assess its usability.

7 CONCLUSION

The presented work on user-driven guidelines for cross-device and IoT authoring tools makes three main contributions. First, we conducted an elicitation study for capturing an end user's mental model when thinking about cross-device and IoT interactions. The presented findings might also serve as an inspiration for future studies that could build on our results and study material to further inform the design and development of XDI and IoT end-user authoring tools. Second, we presented a number of user-driven design guidelines that are mainly based on the analysis of our study results in combination with an investigation of related work. These guidelines form a foundation for the design and development of new XDI and IoT end-user authoring tools as well as for the potential extension and improvement of existing solutions. Finally, we performed a detailed analysis of prominent XDI and IoT end-user authoring tools to check for their compliance with our guidelines, followed by a general discussion about the potential use of our guidelines.

Last but not least, we discussed some initial work on an end-user authoring prototype that is fully compliant with the presented design guidelines. We plan to perform a detailed evaluation of our prototype in combination with discussions in the community—for example based on our classification of existing solutions— which might lead to further refinements and extensions of the presented guidelines and serve as a foundation for the design of future end-user authoring solutions for XDI and IoT applications.

ACKNOWLEDGMENTS

We would like to thank Jonathan Riggio, Reinout Roels and Satyam Kapoor for their help with the user study and the implementation of a first prototype of the authoring tool. The research of Audrey Sanctorum has been funded by a PhD grant of the Research Foundation Flanders (FWO).

REFERENCES

[1] Pierre A. Akiki, Arosha K. Bandara, and Yijun Yu. 2017. Visual Simple Transformations: Empowering End-Users to Wire Internet of Things Objects. *ACM Transactions on Computer-Human Interaction* 24, 2 (April 2017). https://doi.org/10.1145/3057857

[2] Barbara Rita Barricelli and Stefano Valtolina. 2017. A Visual Language and Interactive System for End-User Development of Internet of Things Ecosystems. *Journal of Visual Languages and Computing* 40 (June 2017). https://doi.org/10.1016/j.jvlc.2017.01.004

[3] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Nylandsted Klokmose, and Nicolai Marquardt. 2019. Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices. In *Proceedings of CHI 2019, International Conference on Human Factors in Computing Systems*. Glasgow, United Kingdom. https://doi.org/10.1145/3290605.3300792

[4] John M Carroll, Robert L Mack, and Wendy A Kellogg. 1988. Interface Metaphors and User Interface Design. In *Handbook of Human-Computer Interaction*. Elsevier. https://doi.org/10.1016/b978-0-444-70536-5.50008-7

[5] Juliet M Corbin and Anselm Strauss. 1990. Grounded Theory Research: Procedures, Canons, and Evaluative Criteria. *Qualitative Sociology* 13, 1 (March 1990). https://doi.org/10.1007/BF00988593

[6] Joëlle Coutaz and James L. Crowley. 2016. A First-Person Experience with End-User Development for Smart Homes. *IEEE Pervasive Computing* 15, 2 (April 2016). https://doi.org/10.1109/MPRV.2016.24

[7] Yngve Dahl and Reidar Martin Svendsen. 2011. End-User Composition Interfaces for Smart Environments: A Preliminary Study of Usability Factors. In *Proceedings of DUXU 2011, International Conference on Design, User Experience, and Usability*. Orlando, USA. https://doi.org/10.1007/978-3-642-21708-1_14

[8] José Danado and Fabio Paternò. 2014. Puzzle: A Mobile Application Development Environment Using a Jigsaw Metaphor. *Journal of Visual Languages and Computing* 25, 4 (August 2014). https://doi.org/10.1016/j.jvlc.2014.03.005

[9] Oleg Davidyuk, Iván Sánchez Milara, Ekaterina Gilman, and Jukka Riekki. 2015. An Overview of Interactive Application Composition Approaches. *Open Computer Science* 5, 1 (2015). https://doi.org/10.1515/comp-2015-0007

[10] Rodrigo de A. Maues and Simone Diniz Junqueira Barbosa. 2013. Keep Doing What I Just Did: Automating Smartphones by Demonstration. In *Proceedings of MobileHCI 2013, International Conference on Human-Computer Interaction with Mobile Devices and Services*. Munich, Germany. https://doi.org/10.1145/2493190.2493216

[11] Giuseppe Desolda, Carmelo Ardito, and Maristella Matera. 2017. Empowering End Users to Customize their Smart Environments: Model, Composition Paradigms, and Domain-Specific Tools. *ACM Transactions on Computer-Human Interaction* 24, 2 (April 2017). https://doi.org/10.1145/3057859

[12] Anind K. Dey, Raffay Hamid, Chris Beckmann, Ian Li, and Daniel Hsu. 2004. a CAPpella: Programming by Demonstration of Context-Aware Applications. In *Proceedings of CHI 2004, International Conference on Human Factors in Computing Systems*. Vienna, Austria. https://doi.org/10.1145/985692.985697

[13] Anind K. Dey, Timothy Sohn, Sara Streng, and Justin Kodama. 2006. iCAP: Interactive Prototyping of Context-Aware Applications. In *Proceedings of PERVASIVE 2006, International Conference on Pervasive Computing*. Dublin, Ireland. https://doi.org/10.1007/11748625_16

[14] Daniela Fogli, Matteo Peroni, and Claudia Stefini. 2017. ImAtHome: Making Trigger-Action Programming Easy and Fun. *Journal of Visual Languages and Computing* 42 (October 2017). https://doi.org/10.1016/j.jvlc.2017.08.003

[15] Giuseppe Ghiani, Marco Manca, and Fabio Paternò. 2015. Authoring Context-dependent Cross-Device User Interfaces Based on Trigger/Action Rules. In *Proceedings of MUM 2015, International Conference on Mobile and Ubiquitous Multimedia*. Linz, Austria. https://doi.org/10.1145/2836041.2836073

[16] Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. 2017. Personalization of Context-dependent Applications Through Trigger-Action Rules. *ACM Transactions on Computer-Human Interaction* 24, 2 (April 2017). https://doi.org/10.1145/3057861

[17] Sehyeon Heo, Sungpil Woo, Janggwan Im, and Daeyoung Kim. 2015. IoT-MAP: IoT Mashup Application Platform for the Flexible IoT Ecosystem. In *Proceedings on IOT 2015, International Conference on the Internet of Things*. Seoul, South Korea. https://doi.org/10.1109/IOT.2015.7356561

[18] Jan Humble, Andy Crabtree, Terry Hemmings, Karl-Petter Åkesson, Boriana Koleva, Tom Rodden, and Pär Hansson. 2003. "Playing with the Bits" User-Configuration of Ubiquitous Domestic Environments. In *Proceedings of UbiComp 2003, International Conference on Ubiquitous Computing*. Seattle, USA. https://doi.org/10.1007/978-3-540-39653-6_20

[19] Jeff Johnson. 2010. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. Morgan Kaufmann Publishers Inc., San Francisco, USA.

[20] István Koren and Ralf Klamma. 2016. The Direwolf Inside You: End User Development for Heterogeneous Web of Things Appliances. In *Proceedings of ICWE 2016, International Conference on Web Engineering*. Lugano, Switzerland. https://doi.org/10.1007/978-3-319-38791-8_35

[21] Dejan Kovachev, Dominik Renzel, Petru Nicolaescu, and Ralf Klamma. 2013. DireWolf: Distributing and Migrating User Interfaces for Widget-Based Web Applications. In *Proceedings of ICWE 2013, International Conference on Web Engineering*. Aalborg, Denmark. https://doi.org/10.1007/978-3-642-39200-9_10

[22] Michael Krug, Fabian Wiedemann, and Martin Gaedke. 2014. SmartComposition: A Component-Based Approach for Creating Multi-screen Mashups. In *Proceedings of ICWE 2014, International Conference on Web Engineering*. Toulouse, France. https://doi.org/10.1007/978-3-319-08245-5_14

[23] Thomas Kubitza and Albrecht Schmidt. 2017. meSchup: A Platform for Programming Interconnected Smart Things. *IEEE Computer* 50, 11 (November 2017). https://doi.org/10.1109/MC.2017.4041350

[24] George Lakoff. 1993. The Contemporary Theory of Metaphor. In *Metaphor and Thought*, Andrew Ortony (Ed.). UC Berkeley. https://escholarship.org/uc/item/4nv3j5j9

[25] Toby Jia-Jun Li, Yuanchun Li, Fanglin Chen, and Brad A. Myers. 2017. Programming IoT Devices by Demonstration Using Mobile Apps. In *Proceedings of IS-EUD, International Conference on End-User Development*. Eindhoven, The Netherlands. https://doi.org/10.1007/978-3-319-58735-6_1

[26] Henry Lieberman, Fabio Paternò, and Volker Wulf (Eds.). 2006. *End User Development: An Emerging Paradigm*. Springer. https://doi.org/10.1007/1-4020-5386-X

[27] Jan Meskens, Kris Luyten, and Karin Coninx. 2010. Jelly: A Multi-Device Design Environment for ManagingConsistency Across Devices. In *Proceedings of AVI 2010, International Conference on Advanced Visual Interfaces*. Rome, Italy. https://doi.org/10.1145/1842993.1843044

[28] Brad A. Myers and Mary Beth Rosson. 1992. Survey on User Interface Programming. In *Proceedings of CHI 1992, International Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/142750.142789

[29] Michael Nebeling. 2017. XDBrowser 2.0: Semi-Automatic Generation of Cross-Device Interfaces. In *Proceedings of CHI 2017, International Conference on Human Factors in Computing Systems*. Denver, USA. https://doi.org/10.1145/3025453.3025547

[30] Michael Nebeling and Anind K. Dey. 2016. XDBrowser: User-Defined Cross-Device Web Page Designs. In *Proceedings of CHI 2016, International Conference on Human Factors in Computing Systems*. San Jose, USA. https://doi.org/10.1145/2858036.2858048

[31] Fabio Paternò and Carmen Santoro. 2017. A Design Space for End User Development in the Time of the Internet of Things. In *New Perspectives in End-User Development*, Fabio Paternò and Volker Wulf (Eds.). Springer. https://doi.org/10.1007/978-3-319-60291-2_3

[32] Trevor Pering, Kent Lyons, Roy Want, Mary Murphy-Hoye, Mark Baloga, Paul Noll, Joe Branc, and Nicolas De Benoist. 2010. What Do You Bring To the Table?: Investigations of a Collaborative Workspace. In *Proceedings of UbiComp 2010, International Conference on Ubiquitous Computing*. Copenhagen, Denmark. https://doi.org/10.1145/1864349.1864389

[33] Tom Rodden, Andy Crabtree, Terry Hemmings, Boriana Koleva, Jan Humble, Karl-Petter Åkesson, and Pär Hansson. 2004. Configuring the Ubiquitous Home. In *Proceedings of COOP 2004, International Conference on Cooperative Systems Design, Scenario-Based Design of Collaborative Systems*. Hyères Les Palmiers, France. https://hdl.handle.net/20.500.12015/3031

[34] Mary Beth Rosson and John M. Carroll. 2003. Scenario-based Design. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, Julie A. Jacko and Andrew Sears (Eds.). L. Erlbaum Associates Inc. https://dl.acm.org/doi/10.5555/772072.772137

[35] Luigi De Russis and Fulvio Corno. 2015. HomeRules: A Tangible End-User Programming Interface for Smart Homes. In *Proceedings of CHI 2015, Conference Extended Abstracts on Human Factors in Computing Systems*. Seoul, Republic of Korea. https://doi.org/10.1145/2702613.2732795

[36] Audrey Sanctorum. 2020. *eSPACE: Conceptual Foundations for End-User Authoring of Cross-Device and Internet of Things Applications*. Ph.D. Dissertation. Vrije Universiteit Brussel.

[37] Audrey Sanctorum and Beat Signer. 2016. Towards User-defined Cross-Device Interaction. In *Proceedings of DUI 2016, 5th International Workshop on Distributed User Interfaces*. Lugano, Switzerland. https://doi.org/10.1007/978-3-319-46963-8_17

[38] Audrey Sanctorum and Beat Signer. 2019. A Unifying Reference Framework and Model for Adaptive Distributed Hybrid User Interfaces. In *Proceedings of RCIS 2019, 13th International Conference on Research Challenges in Information Science*. Brussels, Belgium. https://doi.org/10.1109/RCIS.2019.8877048

[39] Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. 2014. Practical Trigger-Action Programming in the Smart Home. In *Proceedings of CHI 2014, International Conference on Human Factors in Computing Systems*. Toronto, Canada. https://doi.org/10.1145/2556288.2557420

[40] Paul Wisner and Dimitris N Kalofonos. 2007. A Framework for End-User Programming of Smart Homes Using Mobile Devices. In *Proceedings of CCNC 2007, International Conference on Consumer Communications and Networking*. Las Vegas, USA. https://doi.org/10.1109/CCNC.2007.146