Intl. Journal of Human–Computer Interaction, 28: 760–767, 2012

ISSN: 1044-7318 print / 1532-7590 online
DOI: 10.1080/10447318.2012.715544

A Comparative Evaluation of User Preferences for Extra-User Interfaces

Jérémie Melchior, Jean Vanderdonckt, and Peter Van Roy
Université catholique de Louvain, Louvain-la-Neuve, Belgium

This article aims to investigate user preferences for extra-user interfaces (extra-UI), formerly known as meta-user interfaces. These are special user interfaces that allow the user to control or personalize the application's user interface. They are named "extra-user interfaces" because the application does not need it to work. Their main goal is the support of systems with several contexts, displays, devices, or platforms. The purposes and features offered by all the Extra-UI vary from one application to another. To create coherence between them, we defined a catalogue of 14 distribution primitives that are typically provided and classified into 4 categories: simple primitives, basic primitives, advanced primitives, and management operations. Based on this catalogue, a comparative analysis of the state of the art was conducted to identify which interaction styles have been properly used and to discuss the rationale behind these usages. From this analysis, we set up and conducted a comparative evaluation of user preferences by 14 participants testing 6 selected distribution primitives in 4 different interaction styles. The research outcomes exemplified that there were significant differences on user preferences between interaction styles with regards to experience level and primitive type.

1. INTRODUCTION

The concept of extra-user interface (extra-UI; Roudaut & Coutaz, 2006), formerly known as Meta-UI (Coutaz, 2006), is defined as the user interface (UI) used for controlling the application UI. An extra-UI may cover a wide spectrum of services ranging from low-level functions such as changing parameters (Melchior, Grolaux, Vanderdonckt, & Van Roy, 2009) to high-level services such as coupling/uncoupling interaction resources (Barralon, 2004). There has been much research on Meta-UI since 2006, but there is no centralization of the results. The direct consequences of the lack of centralization are the use of their own definitions, their own techniques, and their own tools. It is now impossible to find out the right techniques and metaphors to use for extra-UI. Thus, designing and developing an extra-UI today remains an open and challenging research and development problem that is supported by the following motivations that augment existing findings (Vanderhulst, Schrebier, Luyten, Muhlhauser, & Coninx, 2009):

- *Large scope in computer science*: Many different areas of computer science, connected or not, recognize the need for an extra-UI even without naming it as such: ambient intelligence (Balme, Demeure, Barralon, Coutaz, & Calvary, 2004), context-aware computing (Coutaz, 2010), distributed systems (Grolaux, Vanderdonckt, & Van Roy, 2005), multimodal interaction (Benoît et al., 2007; Roscher, Blumendorf, & Albayrak, 2009a, 2009b), pervasive (Vanderhulst, Luyten, & Coninx, 2008) and ubiquitous computing (Hutchings & Pierce, 2006; Paternò, Santoro, Scorcia, Bandelloni, & Mori, 2007; Rekimoto, Ayatsuka, Kohno, & Oba, 2003; Vandervelpen, Vanderhulst, Luyten, & Coninx, 2005), tabletop interaction (Vanacken, Demeure, Luyten, & Coninx, 2008), and in workflow management systems (Biehl et al., 2008).
- *Wide coverage in human activities*: Extra-UIs have been developed for different domains of human activities and objects, thus suggesting that they cover a wide range of interactive applications for these domains: management, electronic commerce, bank and insurance systems, and organization-based systems.
- *Increasing recognition of Extra-UI*: Extra-UIs have been developed since the 1990s but have been significantly recognized since the characterization of this concept in Roudaut and Coutaz (2006). This recognition is not only theoretical by a stable definition of the concept but also empirical by the appearance of new user studies demonstrating its benefits.
- Most of these extra-UIs have been developed without any particular goal
- The extra-UI adheres to the principle "The end user should be kept in the loop" relevant to the area of End-User Development. Developing an extra-UI should result not only from user-centered design (an extra-UI is after all a particular UI) but also from

This research was supported by the European UsiXML ITEA2 Call3 project.

Address correspondence to Jérémie Melchior, Louvain School of Management. Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium. E-mail: jeremie.melchior@uclouvain.be

knowledge coming from other domains, such as distributed systems.

These observations lead us to conclude that there is a common ground between pieces of related work in this area from which a set of generic functions could emerge that would benefit from both a theoretical and empirical analysis that could then be applied and retrofitted. Dearman and Pierce (2008) reported in their U.S. study how 27 people from academic and industrial research were using their platforms: They revealed that on average they employ more than five computing platforms in four different configurations. As soon as different platforms are used by one or different users in different configurations, the problem of coordinating information flow and process across these platforms is posed. Built-in mechanisms exist such as file exchange, file mailing, or file synchronization, but they remain generic, independent from any end-user tasks. To support these users effectively and efficiently, depending on their user tasks, we believe that some facilities could be provided for the end user for sharing, dividing, and assembling the UI portions across these platforms.

To address the aforementioned shortcomings and the problem independently of the application domain as possible, this article provides a comparative evaluation of user preferences for extra-UIs. The remainder of this article is structured as follows: Section 2 delivers a comparative analysis of the state of the art, and section 3 reports on a comparative evaluation of user preferences for interaction styles found for extra-UIs. Section 4 concludes the article by summarizing the main findings and discussing future avenues of this work.

2. RELATED WORK

The first state of the art conducted by Roudaut and Coutaz (2006) is a comparison of 25 extra-UIs in ambient intelligence against criteria representing (a) the objects manipulated by the extra-UI (i.e., their nature and manipulation), (b) the extra-UI external presentation (i.e., embedded or not in the Final User Interface (FUI), offering observability and/or predictability), (c) capability to ensure services (i.e., resource discovery, assembly, distribution, remolding, and parameterizing), and (d) their expandability. This survey revealed that many extra-UIs exist that provide similar or dissimilar capabilities but with very different metaphors (e.g., scissors for cutting, glue for composing) and interaction styles (e.g., form filling, direct manipulation), thus suggesting a further study. Their survey compared high-level services, which suggests that also low-level services could be used as new comparison criteria to complement the survey.

The article (Vanderhulst et al., 2009) pioneered the field by conceptualizing the first reference framework for extra-UIs for pervasive systems. In this framework, an extra-UI runs on *devices* that offer *services* depending on *tasks* executed by *users*. Then environment, the interaction resources, and the domains resources are then characterized and mapped. Based on this framework, we would like to examine the services offered and the way they are offered to determine which aspects are affected by the users carrying out these corresponding tasks.

We have defined and applied a procedure for conducting a comparative analysis of the state of the art to identify which basic services are supported, which interaction styles are used, and for these services in order to discuss the rationale behind these usages. This procedure consists of the following steps:

1. Literature review: All references identified in Roudaut and Coutaz (2006) were initially selected and complemented with related work (only references taken from DataBase systems and logic programming (DBLP)).
2. Classification of references: We listed all basic services found and classified them into four categories of a *catalog of distribution primitives*:

   a. *Simple primitives* [Set, Display, Undisplay, Expose]
   b. *Basic primitives* [Copy, Move, Switch, Permute]
   c. *Advanced primitives* [Merge, Split, Replace, Distribute, Reset, Append]
   d. *Management primitives* [Save, Restore, Undo, Redo, Import, Export]

3. Reduction of scope: The criteria used for selection is any article recognizing the need for distribution of any kind, describing explicitly an implementation, and working at the user interface level (other levels are relevant to distributed computing). A *final comparative analysis* was obtained (Table 1).

In the rest of this section, we only briefly discuss some representative examples issued from this comparative analysis. This discussion helps us to defining the requirements of the comparative evaluation to be conducted in the next section. Here are the possible interaction styles: Drag & Drop, Direct Manipulation, Menu Selection, Multi-Windowing, Iconic Interaction, and Virtual Reality. These interaction styles are further defined and compared in Shneiderman (1988, 1995), Jarke & Vassiliou (1985), and Vanderdonckt (2000).

Table 1 reports on the results of the analytical comparison for the finally selected 12 pieces of work against the possible interaction styles and 17 finally selected distribution primitives. Each distribution primitive is either partially or totally supported. The Multi-Access Service Platform (Roscher, Blumendorf, & Albayrak, 2009a, 2009b) controls distributed user interfaces (DUIs) in smart environments through an extra-UI (Figure 1a) based on a graphical representation of the different tasks that a user can display on a screen. Supported distribution primitives are involved in four services: migration, adaptation, distribution, and multimodality. It allows the distribution of a task on a screen but does not give any choice of the UI that will be displayed. Nevertheless, it focuses on only one single task and has a limited multiuser interaction. There is no support for collaboration, and it is not possible to control the distribution in finer granularity.

Web sites and applications have been investigated with the Migration project (Moscovich, 2009), Cameleon-RT

TABLE 1
Comparative Analysis of Extra-UIs: Their Interaction Styles

Name, reference, and year	Interaction style	Set	Display	Undisplay	Copy	Move	Replace	Transform	Merge	Switch	Separate	Distribute	Reset	Import	Export	Expose	Append	Permute
FormsVBT (1989)	DD, DM, MW, CL	T	T	T		T			P	P							P	
CPN2000 (2000)	PL									P	P	P					P	
Proximal UI (2003)	DD, DM, MW		T	T	T	T	P										P	
Aris (2004)	MW, II					T						P		P	P			
JigSaw (2004)	DD, II		P	P	P	P					P	P					P	P
AttachMe (2005)	II, MS		T	T	T	T				T			T			T		
Lightweight services (2005)	MS		P	P		P			P		P	P					T	
CESAM (2006)	DD, II				P	P	P	P	P		P	P						
MigriXML (2006)	VR, DD, DM		P	P		T						P			T			
Impromptu (2008)	MS		P	P	P	P					P	P			T			
MASP (2009)			P	P		P					P	P						
Ext. Tcl/Tk (2009)	PL	P	T	T	T	T								P	P			

Note. UI = user interface; DD = Drag & Drop; DM = Direct Manipulation; MW = Multi-Windowing; II = Iconic Interaction; MS = Menu Selection; VR = Virtual Reality; MASP = Multi-Access Service Platform; CL = Command Lin; Forms VBT from Avrahami et al. (1989); CPN2000 from Beaudouin-Lafon & Lassen (2000).

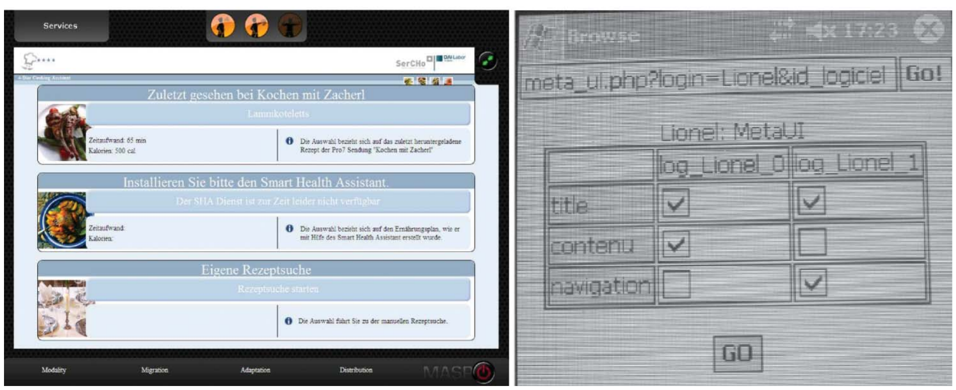

FIG. 1. Some extra-UIs: (a) Multi-Access Service Platform (Roscher, Blumendorf, & Albayrak, 2009); (b) Cameleon-RT (Balme, Demeure, Barralon, Coutaz, & Calvary, 2004) (color figure available online).

(Figure 1b), and Lightweight services (Vandervelpen et al., 2005). There are also window services from one display to another, like copy, move, reshuffle, and rearrange, here based on tabs and metaphors borrowed from Beaudouin-Lafon (2001). Most of the time, the extra-UI is separated from the UI subject to remolding or distribution. For instance, AttachMe (Grolaux et al., 2005) adds icons to the desktop. Some metaphors of extra-UI may be weaved into the UI: A scissor metaphor is used for cutting the window into two regions (Figure 2a). The representation of the world of discourse subject to remolding and distribution is also another property, as in Figure 2b (Biehl & Bailey, 2004). The Drag & Drop of a window from one device to another device results in moving the window to its new position.

IMPROMPTU's (Biehl et al., 2008) extra-UI (Figure 3) supports collaboration between several users with a view on the windows shared by any user (Figure 3a) and view on the

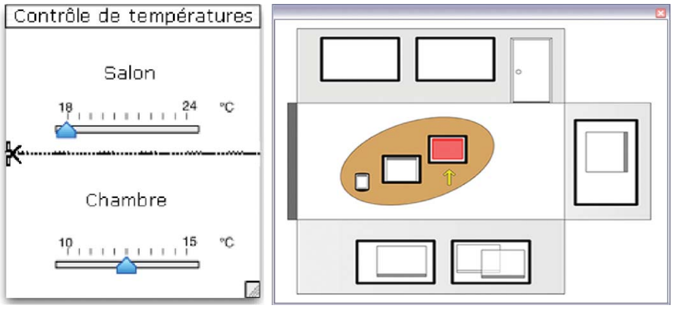

FIG. 2. Advanced extra-UIs: (a) a scissor for cutting; (b) ARIS (Biehl & Bailey, 2004) (color figure available online).

windows shared by a user from the device where it is displayed (Figure 3b). Multiuser collaboration is also supported in the Lightweight Services (Vandervelpen et al., 2005) thanks to a HTTP-based daemon allowing the distribution of web applications. Instead of creating a DUI, distribution of web applications through services offering is provided. MigriXML (Molina, Vanderdonckt, & Gonzáles, 2006) exploits a virtual reality scene obtained from a physical environment model to render windows and dialog boxes of graphical UIs. By pressing a toggle button for migration, the end user moves any window to any display in direct manipulation, which finally results into transferring a UI description from one platform to another.

From this comparative analysis of the state of the art, we could draw the following conclusion: Distribution primitives are implemented in various pieces of work, but in many different ways, with different interaction styles and techniques. We do not know today which one is prevalent depending on the distribution primitive for an extra-UI to become widely acceptable. Little or no work was detected for any meta-UI (according to the definition in section 1) because nothing was found for supporting the manipulation of the abstract UI, the context model, the task model, or the domain model with direct implication on the lower levels.

3. EXPERIMENTAL STUDY

Due to the heterogeneity of distribution primitives and their implementation demonstrated in the previous section, this section aims to conduct an experimental study in order to determine the user preference for particular interaction styles for each distribution primative identified in section 2.

3.1. Method and Protocol

Participants and apparatus. We conducted a user trial of 14 participants (eight female, six male) who were recruited from a database of volunteers coming from different disciplines (e.g., marketing, finance, medicine, management) and having different ages (22% between 19 and 25 years old, 57% between 26 and 40 years old, and 21% older than 40). No participant has any prior knowledge of an extra-UI. The physical setup was similar for all participants. The computer used for the experiment was equipped with an Intel Pentium M 1.6GHz CPU, 2Gb DDR of RAM, and a 15-in. LCD screen with a resolution of 1400 × 1050 pixels. This apparatus was selected because it was considered representative.

Task and procedure. Each participant received a detailed explanation of the experimental study, which was uniformly conveyed through an interactive presentation (implemented with the animation language of Microsoft PowerPoint 2010): to express their preference for interaction styles for distribution primitives. Following the short introduction to the test procedure and test purpose, the 14 participants were presented six distribution primitives (i.e., Set, Copy, Move, Replace, Merge, and Split), each with four different interaction styles for each operator (i.e., A = form filling with iconic interaction, B = direct manipulation with drag & drop, C = command language, and D = menu selection). Each interaction style was implemented as animated examples thanks to Microsoft PowerPoint 2010 macrocommand language. Each participant received a different sequence of interaction styles to avoid the order influencing the results. There are 14 sequences, and each distribution primitive has a different sequence that was generated by a pseudorandom generator. The left of Table 2 shows the order of interaction styles for each sequence, and the right shows which sequence was selected for each distribution primitive for each participant. Each participant therefore provided an answer to the following questions:

- Which devices have you already used or owned? (desktop, mobile phone, tablet PC, interactive kiosk)

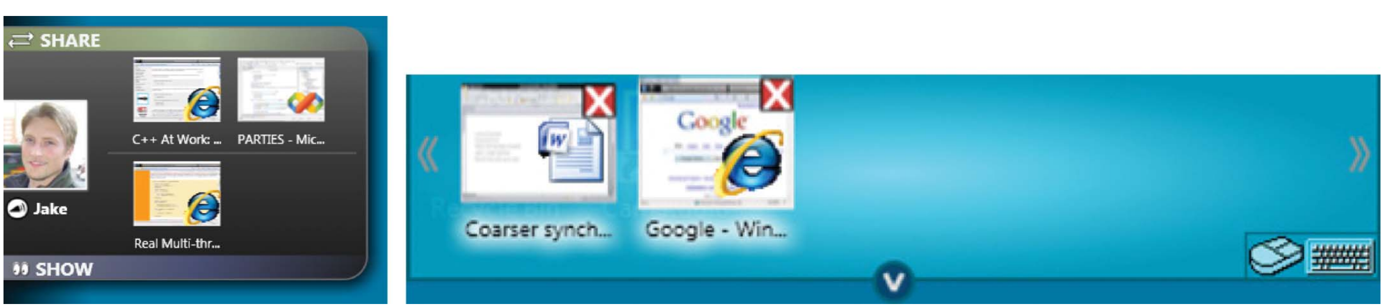

FIG. 3. IMPROMPTU's extra-UI for windows sharing (a) and displaying (b) (color figure available online).

TABLE 2
Random Sequences Generated and Their Repartition by Participant

ID	Random Sequence				Particip.	SET	COPY	MOVE	REPLACE	MERGE	SPLIT
1	C	B	A	D	1	1	2	3	4	5	6
2	A	B	C	D	2	2	3	4	5	6	7
3	A	B	D	C	3	3	4	5	6	7	8
4	C	A	D	B	4	4	5	6	7	8	9
5	B	D	A	C	5	5	6	7	8	9	10
6	D	A	C	B	6	6	7	8	9	10	11
7	B	C	A	D	7	7	8	9	10	11	12
8	B	A	C	D	8	8	9	10	11	12	1
9	A	C	D	B	9	9	10	11	12	1	2
10	B	C	D	A	10	10	11	12	1	2	3
11	D	A	C	B	11	11	12	1	2	3	4
12	D	C	B	A	12	12	1	2	3	4	5
					13	1	12	11	10	9	8
					14	9	8	7	6	5	4

- What is your experience level with a computer/laptop, mobile phone/smartphone, tablet PC, and Interactive kiosk? (preference on a 5-point Likert scale)
- What is your favorite style for each distribution primitive (A, B, C, or D)?
- What are the aspects that you found the most interesting, if any? (open question)
- What are the aspects that you found the less interesting, if any? (open question)
- What do you think of using pen-based gestures for some distribution primitives? (open question)

Two examples of some metaphors that were presented to the participants for the MOVE and MERGE operations are shown in Figure 4. One example of the PowerPoint slides provided to the participants is available; see Evaluation (2011). The dependent variables used to assess the participant task performances were twofold: the preferences for an interaction style for each distribution primitive (i.e., A, B, C, or D) and the comments gathered for each interaction style (i.e., number of positive and negative comments for each interaction style). All the relevant data captured from the presentation were captured in a log file to be imported in a statistical software package. We now justify these main choices of this setup.

Justifications. When it comes to evaluate alternative UI designs, Tullis and Albert (2008) reported that a couple of self-reported metrics are particularly relevant. One is asking each participant to choose which alternative interaction style they would most like to use in the future for a distribution primitive as a forced-choice comparison. Another one is asking each participant to provide comments on each interaction style split into two classes: what were the most positive aspects that you appreciated (if any), and what were the most negative aspects that you regretted (if any). This is what we did. To consistently analyze all verbal comments, Tullis and Albert's protocol for verbal analysis was used, which classifies any comment into three classes: positive (when a clearly positive tone is expressed), negative (when a clearly negative tone is expressed), or neutral (when no clear tone is expressed or in any other case).

The six distribution primitives selected were the most frequently found ones in the literature that were considered fundamental for an extra-UI and because of their associated spectrum of interaction styles. Indeed, we did not find all interaction styles possible for each primitive. SET was the first primitive selected for its simplicity and its largest scope of possible interaction styles. COPY and MOVE were also selected for the same reasons in the set of basic primitives. SWITCH and PERMUTE are less interesting because they could be obtained as a composition of other primitives such as MOVE. The three last operations were REPLACE, MERGE, and SPLIT because of their large coverage and representativeness of the aims and goals of an extra-UI. The DISTRIBUTE and RESET primitives require too much implementation effort, and APPEND primitive is a subprimitive of MERGE, so they were not selected.

The four interaction styles were selected based on their intrinsic properties (Table 3) and because some interaction styles are typically combined with others. For instance, the command language is assumed to be appropriate when the task prerequisites are moderate, the task productivity should be high, and so forth.

3.2. Results and Discussion

The survey was based on a 14 × 6 × 4 factorial design: 14 participants were involved, six distribution operators were selected, and there were four different interaction styles for

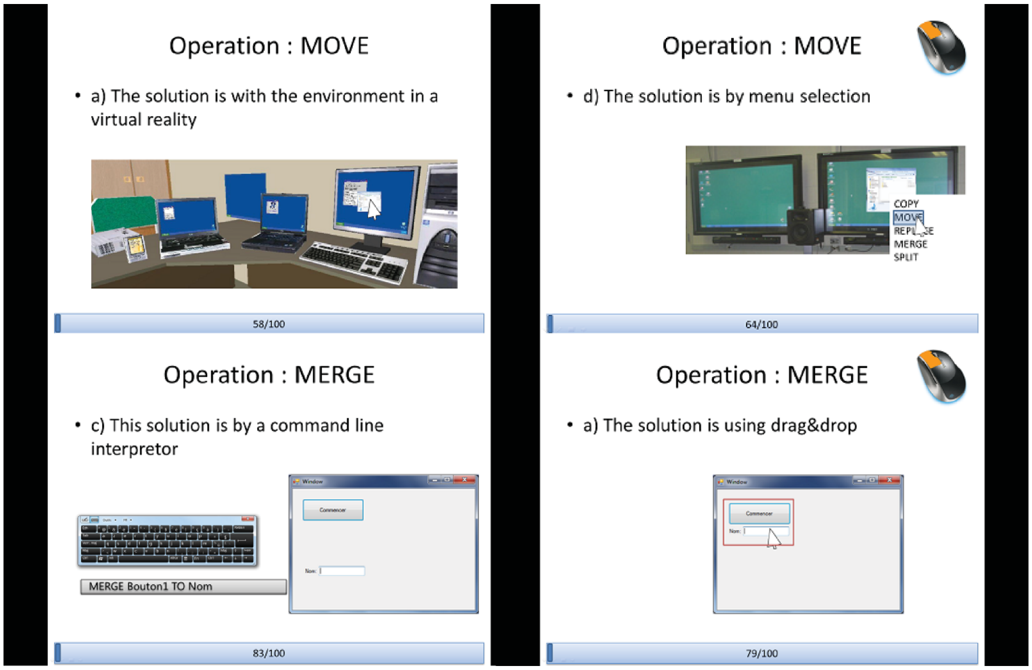

FIG. 4. Example of metaphors for MOVE and MERGE operations (color figure available online).

TABLE 3
Characterization of Interaction Styles in Terms of Task Parameters

Interaction Style	Prerequisites	Productivity	Objective Environment	Task Structuring	Task Importance	Task Complexity
Command language	Moderate	High	Nonexistent	Low	High	Low to moderate
Menu selection	Minimal	Moderate	Nonexistent	Moderate to high	Low	Moderate
Form filling	Moderate	Moderate	Existent	High	High	Moderate
Multi-windowing	Moderate	Moderate	Existent	Low to moderate	High	High
Direct manipulation	Minimal to maximal	Moderate	Existent	Low	High	High
Iconic interaction	Moderate	High	Existent	Moderate	Moderate	Low
Virtual reality	Minimal	Low	Existent	Low to moderate	Low	Low to moderate

each operator. All 14 participants completed the 24 trials, thus giving a total sampling of 336 trials. No outlier was removed because all tasks have been completed without any problem and interruption. Figure 5a summarizes the participants' experiences regarding their usage of various computing platforms, ranging from no experience to expert.

Figure 5b graphically reproduces the results obtained for the user preferences:

- Form filling with iconic interaction (A) was considered globally as the most preferred interaction style for advanced primitives taken together (i.e., Replace, Merge, and Split) in terms of occurrences and a percentile analysis revealed that the 50% percentile (μ_{50}) is in favor of this alternative, followed by menu selection, and direct manipulation.
- Direct manipulation (e.g., with drag and drop; B) is the most preferred interaction style (B) for the Set primitive belonging to the set of simple primitives, probably because of the visual counterpart of the UI element property. This was assessed in terms of occurrences and of the 90% percentile (μ_{90}) which is in favor.

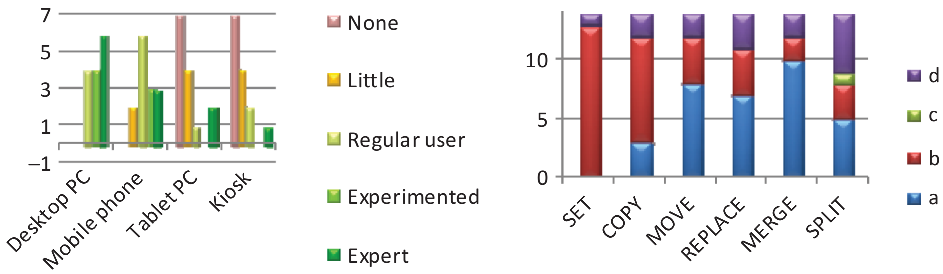

FIG. 5. (a) Platform experience of participants. (b) Distribution of preferred interaction styles for each distribution primitive (color figure available online).

- Direct manipulation is also the preferred interaction style for Copy and Move, belonging to the set of advanced primitives. This was assessed in terms of occurrences and of the 55% percentile (μ_{55}) is in favor of this alternative.
- Command language (C) remains the least preferred interaction style. Command language was only appreciated for the Split primitive because it involves several complex parameters as opposed to more obvious parameters for other primitives.

Figure 6 graphically reproduces the distribution of comments gathered from the participants regarding the overall usage of the six distribution primitives (Figure 6a) and regarding the overall usage of interaction styles for all primitives (Figure 6b). All the tested primitives are considered vital by participants (in terms of occurrences and a 50% percentile [μ_{50}], which is in favor globally speaking). Participants' feedbacks are mostly positive for the primitives. Each primitive has always at least one interaction style that makes it easy to use and natural to understand, particularly Drag & Drop. The styles can vary depending on the context of use, such as the SET operation. There is a question about how precise a primitive like SET should be if we use direct manipulation or Drag & Drop. The diversity of styles proposed for each operation is well appreciated. If the SET has to be precise, they will use direct values by command language, modifying code or changing values of a property. The results for each style taken individually are less mitigated. Although Drag & Drop is widely the most interesting and natural one for the participants, command language and code modification are again the less interesting and less natural. Participants have approved the other styles but seem less convinced by them. The menu selection has been criticized because less intuitive than direct manipulation and Drag & Drop. Virtual reality is very interesting for the MOVE operation. Participants say that it is nice to have Drag & Drop features in virtual reality. It allows the user to see what she is going to do and to specify exactly where she moves the UI elements.

4. CONCLUSION

In this article we have investigated user preferences for extra-UI. The contributions are a catalog of distribution primitives classified into four categories, a comparative analysis of the related work consistent to the catalog, and the first survey on metaphors for extra-UI. The catalog is the first classification of the primitives that can be used for distribution. The outcomes have shown that users have different preferences with regards to the primitives and their experience. The results of the survey highlight the needs of natural metaphors, the importance of feedback, and the difference of preferences. It raises the question of combining the metaphors to let user choose the one he or she prefers depending on the context of use. All the metaphors have been proven to be useful, but they should be adapted to the context, that is, user experience, task experience. Basically, there is still work to do before integrating DUI in a day-to-day environment, and user preferences should not be overlooked.

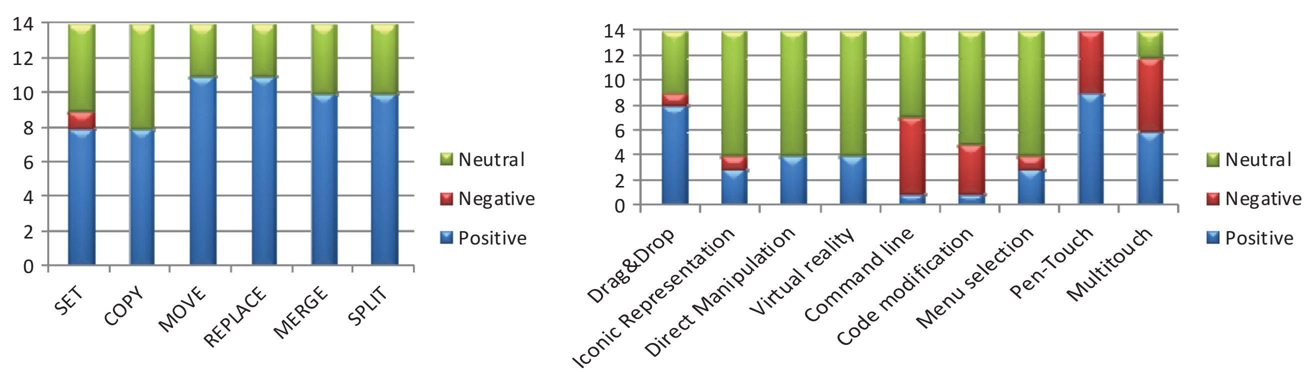

FIG. 6. Distribution of participants' comments for all distribution primitives (a) and all interaction styles, explicit or implicit (b) (color figure available online).

The work is ongoing on a toolkit supporting all the distribution primitives compared in this article. A more complete and complex survey will be realized in order to validate the toolkit.

REFERENCES

Avrahami, G., Brooks, K. P., & Brown, M. H. (1989). A two-view approach to constructing user interfaces. *Proceedings of SIGGRAPH'89*, 137–146.

Balme, L., Demeure, A., Barralon, N., Coutaz, J., & Calvary, G. (2004). Cameleon-RT: A software architecture reference model for distributed, migratable, and plastic user interfaces. *Proceedings of EUSAI'2004*, 291–302.

Barralon, N. (2004). Meta UI: vers un Desktop++. *RJHCI'2004*, pp. 21–24.

Beaudouin-Lafon, M. (2001). Novel interaction techniques for overlapping Windows. *Proceedings of UIST'2001*, 153–154.

Beaudouin-Lafon, M., & Lassen, H. M. (2000). The architecture and implementation of CPN2000, a post-WIMP graphical application. *Proceedings of UIST'2000*, 181–190.

Bandelloni, R., Paternò, F., Santoro, C., & Scorcia, A. (2007). Web user interface migration through different modalities with dynamic device discovery. *Proceedings of AEWSE'2007*, Como, July 2007, pp. 58–72.

Benoît, A., Bonnaud, L., Caplier, A., Jourde, F., Nigay, L., Serrano, M., . . . Lawson, J.-Y. (2007). Multimodal signal processing and interaction for a driving simulator: Component-based architecture. *Journal on Multimodal User Interfaces*, *1*, 49–58.

Biehl, J. T., & Bailey, B. P. (2004). ARIS: An interface for application relocation in an interactive space. *Proceedings of GI'2004*, 107–116.

Biehl, J. T., Baker, W. T., Bailey, B. P., Tan, D. S., Inkpen, K. M., & Czerwinski, M. (2008). IMPROMPTU: A new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. *Proceedings of CHI'2008*, 939–948.

Coutaz, J. (2006). Meta-user interfaces for ambient spaces. *Proceedings of TAMODIA'2006*, 1–15.

Coutaz, J. (2010). UI plasticity: Model driven engineering to the limit! *Proceedings of EICS'2010*, 1–8.

Dearman, D., & Pierce, J. S. (2008). It's on my other computer!: Computing with multiple devices. *Proceedings of CHI'2008*, 767–776.

Evaluation. (2011). Available from https://lilab.isys.ucl.ac.be/groups/lilab/wiki/4eb23/IJHCI_2012.html (last accessed September 18, 2012)

Grolaux, D., Vanderdonckt, J., & Van Roy, P. (2005). Attach me, detach me, assemble me like you work. *Proceedings of RACT'2005*, 198–212.

Hutchings, H. M., & Pierce, J. S. (2006). Understanding the wheters, hows, and whys of divisible interfaces. *Proceedings of AVI'2006*, 274–277.

Jarke, M., & Vassiliou, Y. (1985). A framework for choosing a database query language. *ACM Computing Surveys*, *17*, 313–370.

Melchior, J., Grolaux, D., Vanderdonckt, J., & Van Roy, P. (2009). A toolkit for peer-to-peer distributed user interfaces: concepts, implementation, and applications. *Proceedings of EICS'2009*, 69–78.

Molina, J. P., Vanderdonckt, J., & Gonzáles, P. (2006). Direct manipulation of user interfaces for migration. *Proceedings of IUI'2006*, 140–147.

Moscovich, T. (2009). Contact area interaction with sliding widgets. *Proceedings of UIST'2009*, 13–22.

Rekimoto, Y., Ayatsuka, Y., Kohno, M., & Oba, H. (2003). Proximal interactions: A direct manipulation technique for wireless networking. *Proceedings of INTERACT'2003*, 511–518.

Roscher, D., Blumendorf, M., & Albayrak, S. (2009a). A meta user interface to control multimodal interaction in smart environments. *Proceedings of IUI'2009*, 481–482.

Roscher, D., Blumendorf, M., & Albayrak, S. (2009b). Using meta user interfaces to control multimodal interaction in smart environments. *Proceedings of MDDAUI'2009*.

Roudaut, A., & Coutaz, J. (2006). Meta-IHM, ou comment contrôler son espace interactif ambiant [Meta-IHM, or how to control our ambient interactive space]. *Proceedings of UbiMob'2006*.

Shneiderman, B. (1988). We can design better user interfaces: A review of human-computer interaction styles. *Ergonomics*, *31*, 699–710.

Shneiderman, B. (1995). A taxonomy and rule base for the selection of interaction styles. In R. M. Baecker, J. Grudin, W. A. S. Buxton, & S. Greenberg (Eds.), Readings in human–computer interaction: Toward the year 2000 pp. 401–410. San Francisco, CA: Morgan Kaufmann.

Tullis, T., & Albert, B. (2008). *Measuring the user experience—Collecting, analyzing, and presenting usability metrics*. San Francisco, CA: Morgan Kaufmann.

Vanacken, D., Demeure, A., Luyten, K., & Coninx, K. (2008). Ghosts in the interface: Meta-UI visualizations as guides for multi-touch interaction. *Proceedings of TABLETOP'2008*, 87–90.

Vanderdonckt, J. (2000). A small knowledge-based system for selecting interaction styles. *Proceedings of the International Workshop on Tools for Working with Guidelines, TFWWG'2000*, 247–262.

Vanderhulst, G., Luyten, K., & Coninx, K. (2008). Put the user in control: Ontology-driven meta-level interaction for pervasive environments. *Proceedings of ONTORACT'2008*, 51–56.

Vanderhulst, G., Schrebier, D., Luyten, K., Muhlhauser, M., & Coninx, K. (2009). Edit, inspect and connect your surroundings: A reference framework for meta-UIs. *Proceedings of EICS'2009*, 167–175.

Vandervelpen, C., Vanderhulst, G., Luyten, K., & Coninx, K. (2005). Lightweight distributed web interfaces: Preparing the web for heterogeneous environments. *Proceedings of ICWE'2005*, 197–202.

ABOUT THE AUTHORS

Jérémie Melchior is a Ph.D. Student at Université catholique de Louvain in Belgium. His thesis is about modeling and managing Distributed Systems and Distributed User Interfaces. His main goal is to provide a toolkit that will allow developers to create applications that support Distributed User Interfaces across several operating systems.

Jean Vanderdonckt is the head of Belgian Laboratory of Computer-Human Interaction (BCHI) now renamed as Louvain Interaction Laboratory (LILAB). He is also a professor of Computer Science at the Université catholique de Louvain in Belgium.

Peter Van Roy is a professor of Computer Science at Université catholique de Louvain in Belgium. He is co-author of the book Concepts, techniques, and models of computer programming. He is the co-inventor of the Mozart Programming System and Oz programming language.