

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Mobile Distributed User Interfaces

*Erika Hernández-Rubio, Amilcar Meneses-Viveros  
and Sonia G. Mendoza-Chapa*

## Abstract

The success of a mobile application is due to the usability that the graphical user interface provides. A feature of mobile devices is the limited space for the interaction and the deployment of the graphical user interface. For this reason, user interfaces can have different interaction modalities. However, to work with information that can be complex to display, the use of modalities may not solve this problem. A possible alternative to provide more workspace to the users is through a distributed user interface (DUI). A mobile DUI allows the mobile applications to use two or more devices to execute the user interface. These devices can be Smart TVs or wearable such as smart watches. In this work the concepts of mobile DUI design are discussed, some use cases are presented and it is shown that its development in mobile devices is feasible.

**Keywords:** distributed user interfaces, mobile application, mobile devices, plasticity

## 1. Introduction

In the last years the use of mobile devices has increased. The success of mobile devices is due, among other factors, to its moderate cost, the variety of applications that allow being connected to the Internet, and the ease of use for many of its applications [1, 2]. The usability of the applications of the mobile devices is the main characteristic for the acceptance of the users [2]. This implies that the applications are intuitive and easy to use. To achieve this, researchers and developers have proposed design guides, patterns and templates to achieve applications with good features and easy to use. In addition, due to the diversity of sensors that mobile devices have, they can have different interaction modes and gestures that are used to control applications [3, 4].

Despite all the innovative technological elements for a pleasant user experience presented by mobile devices, for the user they have the restriction of the size of the screen, which reduces their area of work. The range of displays for smartphones is between 4 and 7 inches [5]. The sizes of the tablets are between 7 and 18 inches. And the range for smartwatches is between 1.2 and 2 inches. While there are mobile devices with screens larger than 13 inches, most of these devices are below 10 inches [5].

To get the most out of the work area offered by most mobile devices it is needed to take advantage of the work areas of the different mobile devices that the user has, depending on the context in which the user is. To achieve this, it is possible to design applications for mobile devices that can work with DUIs. That is, an application takes advantage other mobile devices carried by the user or other devices such as Smart TV

in the area where it is located. Another problem in DUIs design is the quality in order to guarantee the usability and functionality of applications that use DUIs [3].

User interfaces have a time component that allows establishing whether the adaptability of their elements will be done dynamically or statically. Dynamic adaptability refers to the changes that the graphical interface makes when the application detects a change of context. Static adaptability is established when the user chooses how the graphical interface will adapt before doing a task or when starting a session. Therefore, several researches have developed concepts such as DUI and plasticity of user interfaces.

In this work we present concepts of DUIs, plasticity, and mobile computing to establish the specific restrictions for DUIs of mobile applications, and to discuss how the plasticity concepts of user interfaces complement the handling of these restrictions to establish the concept of mobile DUI. We present the design methods that have appeared in the literature and emphasize that both are complementary to a mobile DUI design.

## 2. Mobile distributed user interfaces

A mobile DUI is a DUI that takes advantage of mobile devices, communication networks and context of use to distribute user interfaces to take advantage of the display restrictions of mobile devices. It should be clear the concepts of DUIs and the characteristics of mobile applications have a complete notion of mobile DUI.

The use of DUIs is very common in multimedia applications such as music players, video players, image galleries, video games, books or interactive learning materials, but there are still few applications that use it for purposes other than entertainment. DUIs can be used in educational contexts [6] and for assistance applications for disabled persons [6, 7]. Also DUIs are required to interaction with smart spaces [8–11].

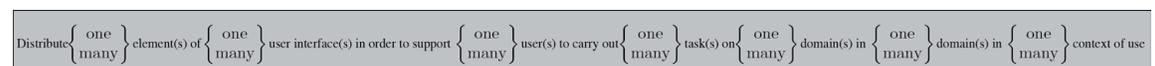
One approach to understand the use of DUIs is in [2], where the authors discuss the evolution of trends of computing since main frames to ubiquitous computing (UC). With the arrival of UC, users interact with more computing devices that contain input and output elements. In this section we start with a discussion of the DUIs, then a discussion of the characteristics of the mobile applications, and finally a discussion is presented to define the concept of mobile DUI.

### 2.1 Distributed user interfaces

A user interface (UI) is the set of elements that allow the user to interact with computers. These elements can be categorized as input, output and data control. This definition involves all kinds of technology and interaction mechanisms.

Vanderdonckt [12] propose a transversal model to distribute the user interface across users, platforms and environments. In this model, the authors consider the triplet  $C = (U, P, E)$ , where  $U$  is the user model,  $P$  is the platform model, and  $E$  is the environment model. Vanderdonckt considers these three elements the dimensions for UI distribution (**Figure 1**).

With this model it is possible to determine what elements of the UI would be distributed, to know the interaction modality that will be used when the elements are ported



**Figure 1.**  
*Transversal model of DUI.*

to target platform, to know the tasks that will be performed in a lapse of time, to know the domains involved in the distribution, and to know the platforms that participate in that distribution configuration.

A distributed user interface is a set of UIs that can be implemented in more than one device, or software platform. Some implementations consider the use of two or more devices simultaneously [7, 13]. By authors Penaver, Melchior and Gallud in several papers from 2011 to 2013 [14–17] we know that any single user interface can be cataloged as a distributed user interface if it has some characteristics like portability, fragmentation (also known as decomposition), simultaneity, and continuity. Being the first two characteristics the most important to satisfied the transformation of a user interface to a distributed one [14, 17, 18].

**Portability.** Means that a user interface can be completely or partially transferred in order to achieve a better user interaction.

**Fragmentation.** Any user interface can be fragmented, only if its different fragments can be run independently without losing functionality.

**Simultaneity.** If a user interface can run in different, software or hardware, platforms and also can be managed at the same time, it means that the UI is a simultaneous system.

**Continuity.** This characteristic is reachable when a system element can be moved to another module. This element is also a part of our distributed user interface, but always preserving its state.

From 2011 to 2013 several authors make some definitions [14–18] to formulate the DUI abstract model that allows developers to arrive at an implementation model. In this model, the elements of interaction (input, output and control), functionality, target, user interface, portability, decomposability, sub-user interfaces, platform, distributed user interfaces, simultaneity, requirements function and concurrency restriction stand out.

### 2.1.1 Definitions

In [7–9], the authors present a mathematical formalization of the DUI to obtain the properties of portability, fragmentation, simulation and continuity. This formalization is based on the next definitions:

**Interaction element:** An interaction element  $e \notin E$  is an element that allows a user  $u$  to make an interaction in a platform  $p$ . An element can be an input element, an output element or a control element.

**Functionality:** Two interactions elements  $e_1, e_2 \notin E$  have the same functionality if a user performs the same action using them.

**Target:** A subset of elements  $E_0 \subset E$  have the same target if  $\forall e_i \in E_0$  a user gets an action of a target task using the element functionality.

**User interface:** An User Interface  $S_0(I)$  is a set of elements that have the same target. From [14, 15], a User Interface is defined by a set of interaction elements that can perform a task in a specific context.

**Platform:** An interaction element  $e \notin E$  exists in a platform  $p \notin P$  if  $e$  is supported, implemented or executed on  $p$ . Furthermore, a user interface  $I$  is supported in  $p$  if  $\forall e \notin I$ , then a user  $u$  can perform an interaction using  $e$  on  $p$ .

**User subinterface:** Let  $I$  be a graphical interface that allows a user  $u$  to reach a target  $T$  on a platform  $p \notin P$ . If the target reached is a subtarget of  $T$ , then the set of elements that is associated with the subtarget form a graphic subinterface.

**Distributed user interface:** A distributed user interface  $DI \in DUI$  is defined as a user interface which has been fragmented and ported. DUI is a set of interaction elements that come from a set of subinterfaces.

**State of user interface:** The state  $S(I)$  of a user interface  $I$  is defined as the set of values or modes associated to the interaction elements and target in a user interface after the user has reached a target associated with  $I$ . Every user interface  $I_i$  has an initial state  $S_0(I)$  that changes when an element of  $I$  is used to make an interaction of  $u$  with  $p$ .

**State of distributed user interface:** The state of a distributed user interface  $S(DI) = (S(I_1), \dots, S(I_n))$  is a  $n$ -tuple where each element is the state of the user interface  $I_i$  that makes up the DUI.

With these definitions it is possible to have a formal description of the characteristics for distributed user interfaces such as portability, fragmentation, simultaneity and continuity. These characteristics are very important for working with distributed user interfaces. Villanueva et al. [3] propose the use of these characteristics as metrics to determine the quality of DUIs.

## 2.2 Mobile applications

A mobile application is an application that runs following the mobile computing paradigm. In this paradigm, the application's view layer runs on a mobile device, and the business and storage layers may or may not be on or off the device. In addition, the device must have the ability to be always connected (anywhere at any time) taking advantage of the different infrastructures of communication networks and also must consider the mobility of the user [19]. Mobileness means that the use of an application is always under an environment with constant changes, so the application must be able to adapt to changes in the context to remain functional and usable to the end user.

A mobile application is a computer mobile software designed to perform a task or to provide a user experience. The mobile software development presents some special requirements [20]:

**Interaction with other applications:** most of the mobile devices have many applications from different sources. New applications should be able to interact with the installed applications.

**Sensor handling:** the applications must be able to use the device sensors in order to improve user experience.

**Families of hardware and software platforms:** most embedded devices execute code that is custom-built for the properties of that device, but mobile devices may have to support applications that were written for all of the varied devices supporting the operating system, and also for different versions of the operating system.

**User interfaces:** they must be usable. The design of a user interface must consider the device's constraint like display size, battery life and processor capacity, and it must take advantage of the device's capabilities.

**Power consumption:** many aspects of applications affect the use of the device's power and thus the battery life of the device. Mobile applications may make extensive use of battery.

## 2.3 Plasticity of user interfaces

Techniques for reconfiguring the components of an application must be used. In addition to these concepts, it should be considered that a user interface in a mobile environment would be affected by changes in the environment, so the term plasticity turns out to be relevant. The plasticity of user interfaces is their ability to adapt to the context of use and to preserve their usability [21, 22]. This concept is useful to handle the adaptation of the elements in a DUI. Due to the mobility and ubiquity inherent in this type of systems, the changes of context are natural in this type of systems [23].

The context deals with the evolution, the structuring and the exchange of information spaces [24], which are designed to fulfill a particular purpose. In plastic user interfaces, the purpose is to support the process of adapting the user interface to preserve usability, i.e., plasticity techniques must handle the context of use. A change of context could be defined as the modification of any element of the contextual information space.

Vanderdonckt et al. [23] define seven dimensions to manage plasticity: adaptation means, UI component granularity, state recovery granularity, UI deployment, context of use, technological space coverage, and plastic meta-UI.

## 2.4 Mobile DUI

In the literature there are several models to design a DUI and give quality. We can notice that these models complement each other. Vanderdonckt's transversal model uses three dimensions, Penalver's, Melchoir's and Gallud's model uses four dimensions, and Vanderdonckt's plastic model considers seven dimensions. The work that has been done with DUIs and their formalization establish that the elements of a UI can be distributed, and the relationship that exists between them. Those works on plasticity of user interfaces establish how adaptability can be made, focusing on the conditions of context.

To handle the context for a mobile DUI, it must be considered that the information spaces are the elements of the UI (elements, sub UI, etc.), and the characteristics of the devices where the DUI will be displayed. The plasticity of the UI must handle the context of use. A change of context is the set of devices where elements of the user interface can be displayed, and in this way it is observed that elements of the information space are modified.

In general, it is possible to distinguish two methods for DUI design. One of these is presented by [12] and the other is presented by [14–17]. In [12] the way to distribute the GUI elements between users, platforms and environments is emphasized. In [14–16] the design is considered through a conceptual model based on portability, decomposition, simultaneity, and continuity of the DUI. The formalization of the DUI helps to know what elements are going to be distributed. Plasticity helps to establish when to make the distribution of elements and also raises the problem of how to do it.

We can say that a mobile DUI is a set of DUIs that mainly uses mobile devices sensitive to context, which can be supported by ubiquitous computing. The use of mobile devices allows the use of the different interaction modalities that they include. However, DUIs must consider handling the restrictions of mobile devices, mainly the size associated with the user interface, the high dependence on network connectivity and battery management.

There are several mobile platforms such as Android, iOS or Windows Mobile, among others, that provide tools and frameworks for developing applications. Furthermore, kits and frameworks have been developed to create mobile applications that allow us to share displays among mobile devices of the same family, i.e., their frameworks allow us to build some DUIs. Every platform uses a different strategy because the development paradigm is different for each platform. Another option is to develop rich clients that run on a Web browser.

However, despite the advantages offered by mobile application development platforms, it is necessary to use middleware and frameworks that help to efficiently manage DUIs. Another problem remains the design of the UIs that will be distributed to each platform involved in the DUI.

Work has been done to have models for the design, development and deployment of DUIs in execution time [8, 14, 25–28]. These works consider software engineering

techniques as well as aspects of implementation. These last considerations can be reinforced with the works on plasticity that have been developed [29, 30].

### 3. Examples

With the elements described in the model of Section 4, we present three examples where the design of the DUI is available in three combinations of computing devices: Tablet-Smart TV, smartphone-smartwatch, and Tablet-Tablet.

#### 3.1 Tablet-SmartTV

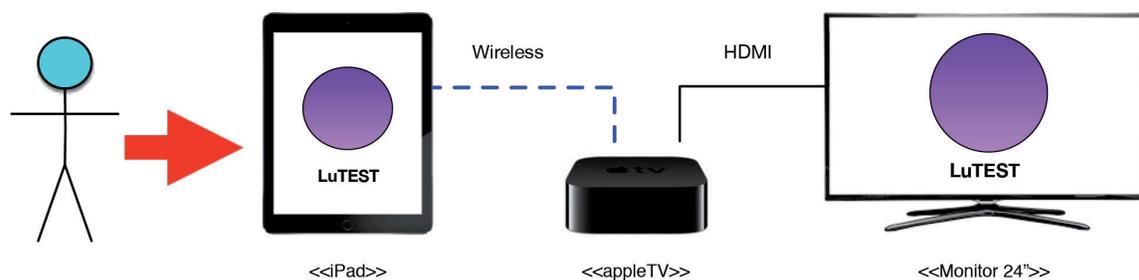
In this example, an application is presented to perform three neuropsychology tests of Luria: Poppelreuter I, Poppelreuter II and Raven [7]. This application is called LuTest, whose architecture allows the user to manage a DUI whose platforms are an iPad tablet and an Apple TV, as shown in **Figure 2**. The main task of this application is to apply to a user the neuropsychological tests of Poppelreuter I, II and Raven. This example presents two ways to show the elements of the output: one is to duplicate the UI and the other to divide the UI. The final users of these applications are older adults, so the design of the UI is aimed at this population. Applications can only run on the Tablet or they can use the Tablet-Smart-TV combination in order to increase the work area.

DUIs have a static adaptability. The user determines the tablet orientation and the mode of work: alone or with a Smart TV before starting the test. The designs of the UI, in all cases of adaptability, are oriented to work with older adults. Because a dynamic adaptability could generate confusion in final users, we decide make a static adaptability.

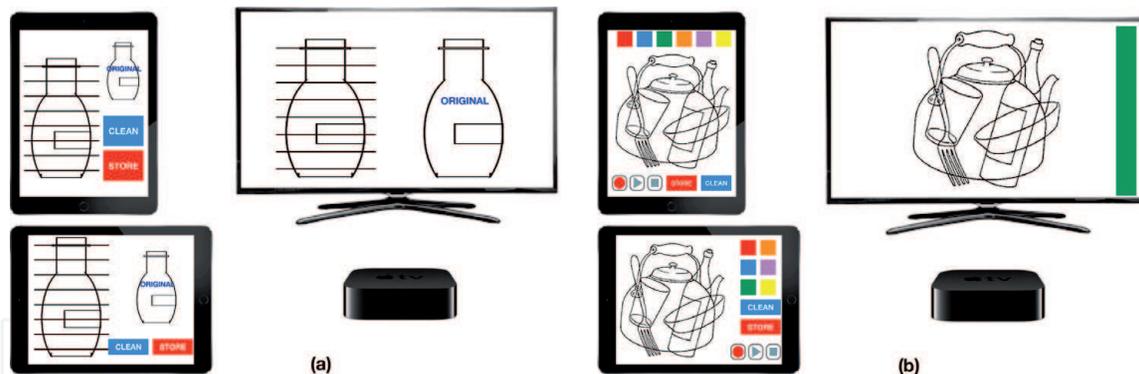
##### 3.1.1 DUI properties for Poppelreuter I and II

In Poppelreuter I test, the user must indicate the figures presented to them with visual noise. Poppelreuter 1 test begins by showing the user the contour image of an object, later more images containing the original object will be shown, but now the outline is combined with lines that may confuse the patient. The user must indicate the outline of the original object, ignoring the additional lines. The test consists in displaying different images, with different objects and different types of visual noise, as shown in **Figure 2**. For Poppelreuter II test, the visual noise is generated by overlapping the contours of several forms, **Figures 2 and 3**. The user must indicate the contour for each of them.

**Portability:** In Poppelreuter I and II there is partial portability. In the tablet the UI is maintained and in the Smart TV the output elements are deployed. These elements display the contours of the figures. In addition, the user can notice the progress of the test and the type of color they have selected to identify each figure separately.



**Figure 2.** Architecture for LuTEST. This application makes a DUI using a Tablet iPad and an Apple TV.



**Figure 3.**  
(a) DUI design for Popperreuter 1 test using a tablet in landscape or portrait orientation with a Smart TV and  
(b) DUI design for Popperreuter II test using a tablet in landscape or portrait orientation with a Smart TV.

**Fragmentation:** The initial UI is divided into two UIs. The UI 1 contains the elements that the contour figures display and allows interaction through the touch screen. The UI2 contains the elements such as buttons and color palettes that the user can choose to perform the test.

**Simultaneity:** When the user works with the input elements found in the UIs of the Tablet, the status changes are reflected on the Smart TV in real time.

**Continuity:** This property is not essential in this example, because the UI enters an initial state when distributing the user interface. Due to the requirements of the application, the elements of the user interfaces do not move during the application of the neuropsychological tests.

### 3.1.2 DUI properties for Raven test

The Raven test is used to evaluate visual and cognitive abilities. It works as follows: the patient observes a certain visual structure, which is incomplete. The patient can choose between six or eight possible options, but only one is correct. In some cases, the patient is asked to differentiate their answers from the others, and for that the patient must grasp the principle under which each option was constructed. The complete Raven test is composed of three series, each with 12 different test matrices whose difficulty progresses step by step. The advantage of using Raven to assess cognitive abilities is that a grammatical knowledge or a complex mathematical ability is not required (**Figure 4**).

**Portability:** The UI is partially transferred from the Tablet to the Smart TV. The input elements are maintained in the Tablet and the output element is sent to the Smart TV.

**Fragmentation:** The UI is fragmented in two sub UI. The UI1 has all input elements. The UI2 has the output element.

**Simultaneity:** When the user works with the input elements found in the UIs of the Tablet, the status changes are reflected on the Smart TV in real time.

**Continuity:** This property is not essential in this example, because the UI enters an initial state when distributing the user interface. Due to the requirements of the application, the elements of the user interfaces do not move during the application of the neuropsychological tests.

## 3.2 Smartphone-smartwatch

In this example, a DUI is presented, which allows the communication of smartwatch and smartphone type mobile devices. The objective of this DUI is to show the best walking route that a tourist should follow in the Historic Center of



**Figure 4.**  
*DUI design for Raven test using the Tablet in landscape or portrait orientation with the Smart TV.*

Mexico City to reach a point of interest around it, in a radius no greater than 5 km. The user can execute the application on the smartphone, on the smartwatch or in both devices using a DUI.

The user decides at any time to activate the DUI, from the smartphone or from the smartwatch. If the user activates the DUI from the smartphone and the smartwatch does not have the application active, then the application is activated and the smartphone sends the status of the application, which indicates that it is in some search of a site of interest or that it is displaying geographic information. In this case, the smartphone starts the application. If the user activates the DUI from the smartwatch and the application is not active on the smartphone, then the status of the application is activated and transferred, so that the smartphone knows the activity that it must present on the display.

**Figure 5** shows the DUI for the search and guide application of sites of interest. The DUI uses the deployment area of the smartphone and smartwatch. In the case of search by predetermined sites, a list is presented on both devices. To guide the user to the site of interest, the smartphone presents the route on a map and an arrow indicating where to go. To guide the user to a site of interest, the smartwatch presents an arrow indicating the orientation.

**Portability:** Depending on the state of the application the UI is duplicated in both devices (for example, search for interest sites) or part of the UI is displayed on the smartwatch and the UI complete is displayed on smartphone.

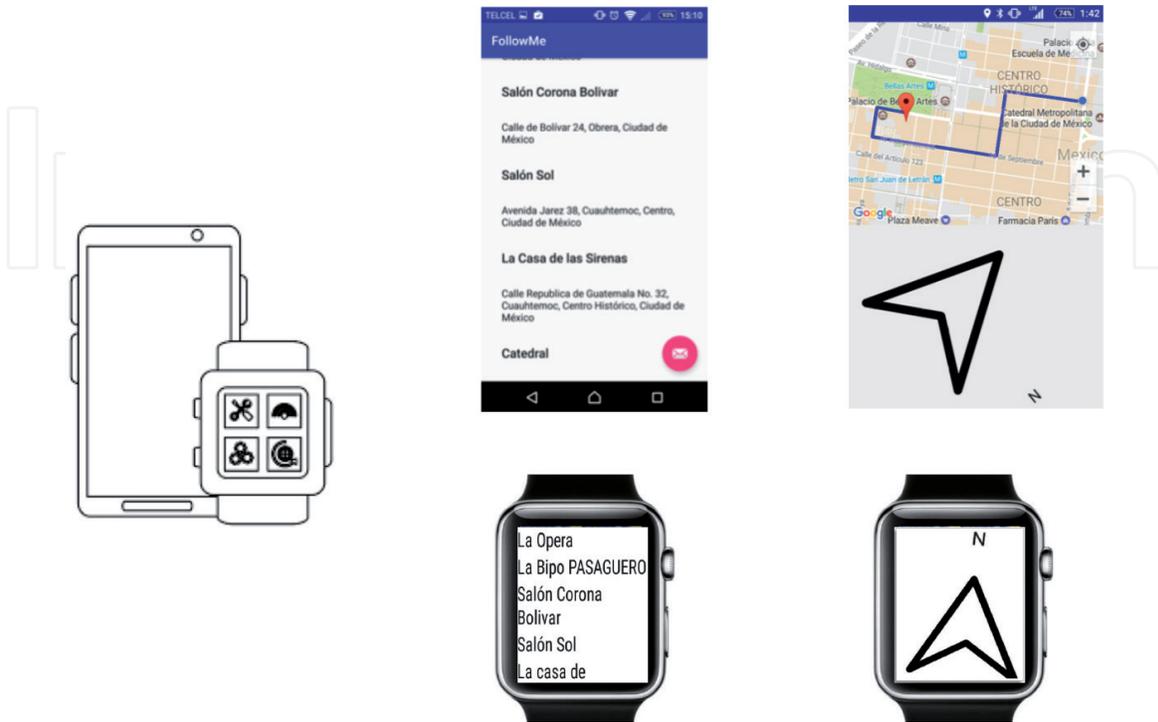
**Fragmentation:** Several screens of the application are duplicated in the smartphone and in the smartwatch, for these cases there is not fragmentation. When the application displays the map to indicate the route to the user to reach a site of interest, the UI is fragmented into two elements: one element E1 displays the map and the element E2 displays the date indicating the orientation of the site of interest. The smartphone displays E1 and E2 and the smartwatch only displays E2.

**Simultaneity:** Both devices display in real time the changes made by the user.

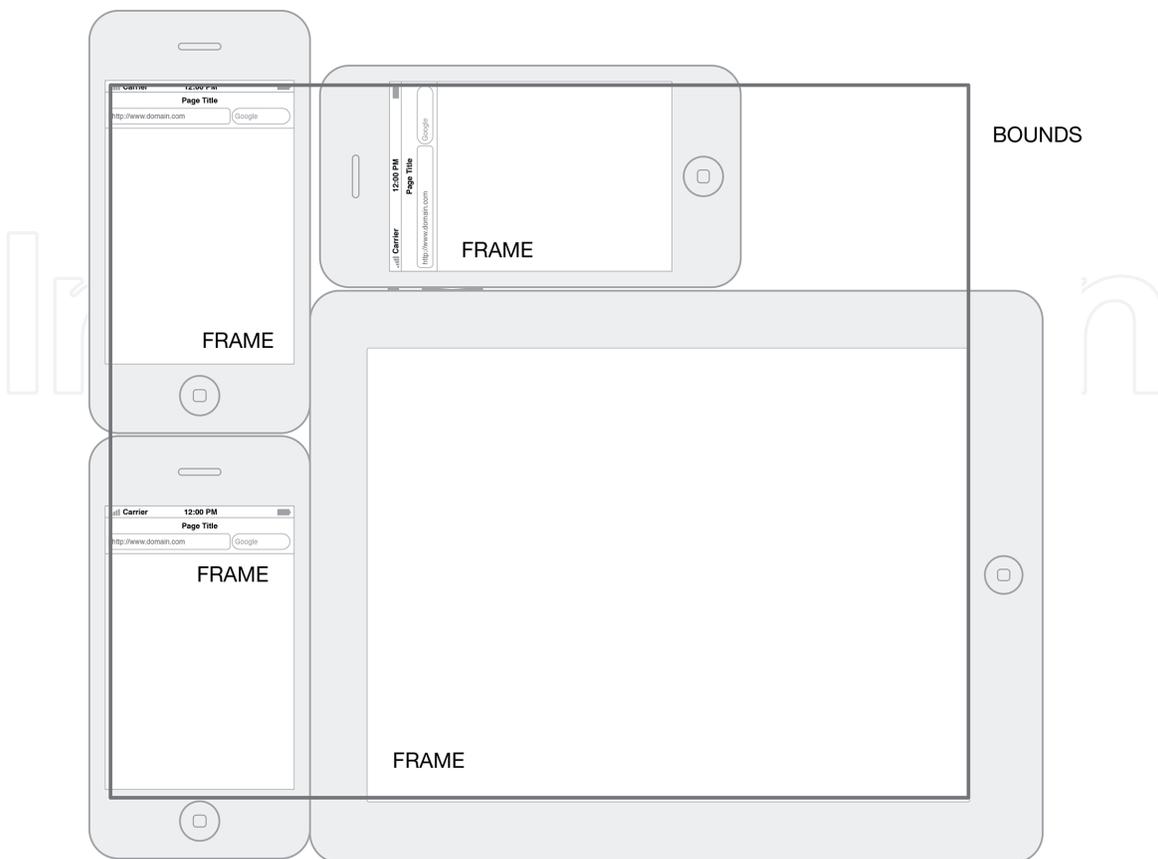
**Continuity:** The user defines when starting using the DUI and when finishing. When the user decides to activate the DUI and the application in smartphone is synchronized with the application in the smartwatch.

### 3.3 Tablets-smartphones

This example uses a set of tablets or smartphones to increase the working area. The DUI is increased dynamically when the application detects another device with the application. The application is a mental maps editor. The application detects



**Figure 5.**  
*DUI design for an application using smartphone and smartwatch.*



**Figure 6.**  
*DUI design for application using smartphones and tablets.*

a gesture to add or remove a device from the application, and therefore adjust the DUI dynamically (**Figure 6**).

**Portability:** The UI has one element. This element is a canvas to draw and mental map. When a device is incorporated to the DUI, the element is duplicated in another device.

**Fragmentation:** Every display in the DUI display a part of general working area. Each device displays a part of the canvas. The canvas has a general work area called Bounds. And every device display a subarea called Frame. The Bounds is increases as devices are added.

**Simultaneity:** All devices must handle the changes in the Bounds. In addition to adding elements to the editor, this change of state of the canvas is sent to all the devices in the array. All devices must handle the changes in the Bounds. In addition to adding elements to the editor, this change of state of the canvas is sent to all the devices in the array. Thus, adding, removing or modifying canvas elements generates a message sending to the devices to update the state of the objects that are drawn on the canvas.

**Continuity:** When devices are added to the array, the canvas state is transferred to the new device and display one part of canvas.

## 4. Conclusions

The trends in DUIs its about the real time system for make the distributions; have software engineering methodologies for the design and implementation of DUIs; have consistent development frameworks and effectively incorporate the context management of applications and users. In this chapter the concept of mobile distributed user interface was made. This concept is based on models of distributed users interfaces, plasticity of user interfaces and mobile applications concepts. The concepts for DUIs help to indicate about elements and sub UIs that can be distributed and define the platforms host. The plasticity of user interface indicates when the applications must fragment de UI, depending on the context state. Now, the main issue in mobile distributed user interface is to decide how to adapt efficiently the sub user interfaces on its platform target. In this work we present some examples of mobile DUIs. We notice that the adaptability of sub user interfaces depends on the user interaction requirements with the application, that include the user group, the device target, and the elements of sub user interfaces, among others, as suggested in [28]. The way to distribute sub user interfaces depends on the application and the devices considers in their use. In some cases it is necessary duplicate elements to others devices but only for output, remaining the input in the original devices, such as the case Tablet-Smart TV, where the Tablet remains the user input, but the output is reply in both devices. In other cases, de GUI is decomposed and then, the input elements remain on the Tablet and the outputs elements are sent to Smart TV. In these cases the input interaction is always on the tablet. For the case of the tourist guide using a smart phone and smart watch, consider several scenarios depending on the user cases.

## Acknowledgements

Authors should like to thank to Cinvestav-IPN and Instituto Politécnico Nacional, SIP project number 20196705 “Diseño de la arquitectura de middleware para protocolos criptográficos en dispositivos restringidos” by the resources provided and the facilities for this work.

IntechOpen

IntechOpen

### **Author details**

Erika Hernández-Rubio<sup>1\*</sup>, Amilcar Meneses-Viveros<sup>2</sup> and Sonia G. Mendoza-Chapa<sup>2</sup>

1 Instituto Politécnico Nacional, SEPI-ESCOM, Mexico City, Mexico

2 Computer Science Department, Cinvestav-IPN, Mexico City, Mexico

\*Address all correspondence to: [ehernandezru@ipn.mx](mailto:ehernandezru@ipn.mx)

### **IntechOpen**

---

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Rashedul I, Islam R, Mazumder T. Mobile application and its global impact. *International Journal of Engineering & Technology (IJEST)*. 2010;**10**:6:72-78
- [2] Tesoriero R, Gallud JA, Lozano MD, Penichet VM, Vanderdonckt J. Distributed user interfaces: Collaboration and usability. In: *CHI'12 Extended Abstract on Human Factors in Computing Systems*; 5-10 May 2012; Austin, Texas: ACM; 2012. pp. 2719-2722
- [3] Villanueva PG, Tesoriero R, Gallud JA. Is the quality in use model valid for distributed user interfaces. In: *Proceedings of the 2nd Workshop on Distributed User Interfaces: Collaboration and Usability (DUI 2012)*; 5-10 May 2012; Austin, Texas: ACM; 2012. pp. 39-44
- [4] Cutugno F, Leano VA, Rinaldi R, Mignini G. Multimodal framework for mobile interaction. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*; 21-25 May; Capri Island, Italy: ACM; 2012. pp. 197-203
- [5] Raptis D, Tselios N, Kjeldskov J, Skov MB. Does size matter?: Investigating the impact of mobile phone screen size on users' perceived usability, effectiveness and efficiency. In: *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM; 2013. pp. 127-136
- [6] Leite FLV, Prietch SS, Preti JPD. Empowerment of assistive technologies with mobile devices in a DUI ecosystem. *Procedia Computer Science*. 2015;**67**:358-365
- [7] Caballero PC et al. Distributed user interfaces for Poppelreuters and Raven visual tests. In: *International Conference on Human Aspects of IT for the Aged Population*. Cham: Springer; 2017. pp. 325-338
- [8] Luyten K, Coninx K. Distributed user interface elements to support smart interaction spaces. In: *Seventh IEEE International Symposium on Multimedia (ISM'05)*; IEEE; 2005. p. 8
- [9] Gallud JA, Penichet VM. Distributed user interfaces: Distributing interactions to facilitate universal access. *Universal Access in the Information Society*. 2017:1-2
- [10] Tesoreiro R, Altalhi AH. Model-based development of distributable user interfaces. *Universal Access in the Information Society*. 2017:1-28
- [11] Sanctorum A, Signer B. Towards end-user development of distributed user interfaces. *Universal Access in the Information Society*. 2017:1-15
- [12] Vanderdonckt J. Distributed user interfaces: How to distribute user interface elements across users, platforms, and environments. In: *Proceedings of XI Interacción*; 2010. pp. 20
- [13] Sjölund M, Larsson A, Berlung E. Smartphone views: Building multi-device distributed user interface. In: *Mobile HCI 2004*; Springer; 2004. pp. 507-511
- [14] Antonio P et al. Defining distribution constraints in distributed user interfaces. *Journal of Universal Computer Science*. 2013;**19**(6):831-850
- [15] Melchior J, Vanderdonckt J, Van Roy P. A model-based approach for distributed user interfaces. In: *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*; ACM; 2011. pp. 11-20

- [16] Melchior J, Vanderdonckt J, Van Roy P. Distribution primitives for distributed user interfaces. In: *Distributed User Interfaces*. London: Springer; 2011. pp. 23-31
- [17] Gallud JA et al. A proposal to validate the user's goal in distributed user interfaces. *International Journal of Human Computer Interaction*. 2012;28(11):700-708
- [18] Peñalvert A et al. Distributed user interfaces: Specification of essential properties. In: *Distributed User Interfaces*. London: Springer; 2011. pp. 13-21
- [19] Hernandez IMT, Viveros AM, Rubio EH. Analysis for the design of open applications on mobile devices. In: *Proceedings of CONIELECOMP 2013, 23rd International Conference on Electronics, Communications and Computing*; IEEE; 2013. pp. 126-131
- [20] Wasserman T. Software engineering issues for mobile application development. In: *FoSER 2010*; 2010
- [21] Thevenin D, Coutaz J. Adaptation and plasticity of user interfaces. In: *Workshop on Adaptive Design of Interactive Multimedia Presentations for Mobile Users*; 1999. pp. 7-10
- [22] Coutaz J, Calvary G. HCI and software engineering for user interface plasticity. *Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. 3rd ed. CRC Press; 2012. pp. 1195-1220
- [23] Vanderdonckt J et al. Multimodality for plastic user interfaces: Models, methods, and principles. In: *Multimodal User Interfaces*. Berlin, Heidelberg: Springer; 2008. pp. 61-84
- [24] Winograd T. Architectures for context. *Human Computer Interaction*. 2001;16(2-4):401-419
- [25] Coutaz J. User interface plasticity: Model driven engineering to the limit. In: *Engineering Interactive Computing Systems (EICS 2010) International Conference*; Berlin Germany: ACM; 2010. pp. 1-8
- [26] Sottet JS, Calvary G, Favre JM. Models at run-time for sustaining user interface plasticity. In: *Models@ run-time workshop*; 2006
- [27] Sottet JS, Ganneau V, Calvary G, Coutaz J, Demeure A, Gavre JM, et al. Model-driven adaptation for plastic user interfaces. In: *IFIP Conference on Human Computer Interaction*; Heidelberg Berlin: Springer; 2007. pp. 397-410
- [28] Sottet JS, Calvary G, Coutaz J, Favre JM. A model-driven engineering approach for the usability of plastic user interfaces. In: *IFIP Conference on Human Computer Interaction*; Heidelberg Berlin: Springer; 2008. pp. 140-157
- [29] Calvary G, Coutaz J, Thevenin D. Supporting context changes for plastic user interfaces: A process and a mechanism. In: Blandford A, Vanderdonckt J, Gray P, editors. *People and Computers XV—Interaction without Frontiers*. London: Springer; 2001. pp. 349-363
- [30] Thevenin D, Coutaz G. Plasticity of user interfaces: Framework and research agenda. In: *Human-Computer Interaction (INTERACT'99)*. Edinburgh: IOS Press; 1999. pp. 110-117