Distributing Interaction in Responsive Cross-Device Applications

Félix Albertos-Marco, Víctor M.R. Penichet, and José A. Gallud

University of Castilla-La Mancha,
Computer System Department Albacete, Spain
{felix.albertos,victor.penichet,jose.gallud}@uclm.es

Abstract. With the emergence of the Internet of Things there are many applications where the interaction is distributed over multiple devices. Developing applications on these scenarios is challenging because there is not enough knowledge and even less consensus on how to distribute interaction. But following the ongoing trends such as responsive web design, why not enable applications to seamlessly adapt interaction to take advantage of available devices at any moment? In this work we introduce the foundations of a new approach called Responsive Cross-Device Applications (RCDA). RCDA applies the idea of responsive Web applications distributing user interactions into the new cross-device ecosystem, taking into account interactive capacities of devices and users.

Keywords: Responsive Interaction, Cross-Device Applications, Distributed Interaction, Interactive Capacities.

1 Introduction

Distributed Interaction refers to the interaction process where user/s use/s dynamically separated input/output devices (e.g. document viewer [5] [6] , snake-like game [5], exploring a distributed map [6] [2] or video streaming applications [5] [6] [2]). With the emergence of the Internet of Things, there is a promise about a future in which devices can be always connected. As a result, developing applications for these scenarios is challenging because there is not enough knowledge and even less consensus on how to distribute interaction.

On those scenarios, Santosa [4] identified specialization as the main characteristic to improve multi-device interaction and coordination of devices. Hamilton et. al [2] found that largely unexplored are interactions which enable chaining of device functionality and cross-device experience found to be painfully absent in today's technologies. Yang presented Panelrama [6], a web-based framework for the construction of applications using distributed user interfaces. It allows users to interact with a single application from multiple devices that dynamically change user interface allocation to best-fit devices. Jokela [3] found that users

want to use their devices to seamlessly work with each other, but in practice they continuously encountered problems in multi-device use. Schreiner presented Connichiwa [5], a framework for the development of cross-device web applications focused on the integration of existent devices, independent from the network infrastructure, versatility over application scenarios and API usability. Chi et al. presented Weave [1] a framework for developers to create cross-device wearable interaction by scripting.

Most of the developments are ad-hoc solutions based on particular case studies and/or developers intuition. They do not fully take into account the whole interactive system: users, devices and tasks. There is a lack of consensus of how to distribute interaction according to the capabilities of the interactive system and their relation to the main goal of every interactive system: tasks accomplishment. But, following the ongoing trends such as responsive web design, why not enable applications to seamlessly adapt interaction to take advantage of available devices at any moment? This work provides the foundations for a new approach aimed to deal with the design and development of Responsive Cross-Device Applications (RCDA) in nowadays cross-device applications. The idea applies responsive Web applications distributing user interactions into the new cross-device ecosystem.

The paper is structured as follows. First, the two pillars of RCDA are presented. Next, the foundations of our proposal are described: users and devices characterization, adaptations responsibility, runtime adaptations and user patterns. Finally, conclusions and future work are presented.

2 Responsive Cross-Device Applications

The new approach of RCDA has two pillars: Responsive Web applications, and Cross-device applications. Next paragraphs explain these two foundation ideas.

Nowadays, and moreover in the future, seems to be clear that applications must support user's task over multiple devices. This tendency has some similitude with the Web and how Web applications development and use evolved to be what they are now: Responsive Web applications. They adapt to some characteristics of devices such as size, screen resolution, orientation or input modalities. But today's environments are far more complicated.

In a similar way, the coordination of tasks in Cross-device applications is complex [1]. Current approaches do not allow for simultaneous use of devices or Cross-device interaction. Suitable applications are needed to deal with the wide range of existent cross-device scenarios [5].

However, responsive design in Cross-device applications should also considers the fact that applications distribute functionality among different devices. Then, new features regarding such a distribution should be applied in the design. Therefore, following the aforementioned two pillars, we propose a new approach for the design of Cross-device applications: Responsive Cross-Device Applications (RCDA). RCDA adapts the interaction according to the ecosystem, which is composed by devices, users and tasks. The main goal of RCDA is to support

users' tasks in cross-device environments adapting interaction to easy users' tasks completion.

From the analysis of previous works we set the foundations for the design and development of RCDA. The foundations are the following:

1. Users and devices characterization.
2. Adaptations responsibility.
3. Runtime adaptations.
4. User patterns.

These foundations are described in the following sections.

2.1 Users and Devices Characterization

Users and devices characterization has to take into account not only properties such as size and display resolution, but also interactive capabilities of users and devices, allocation of specific roles of each device and/or actor, coordination of devices for simultaneous use and determine what device is best suited for.

In previous approaches, the description and categorization of interactive capabilities has been limited to physical properties such as size and display resolution. Only a few works [6] used other devices capabilities. But these descriptions are limited to concrete parts of the user interface and some characteristics of devices' capabilities.

In our approach the description of the interactive system is based on the interactive capabilities of the actors involved in the interaction process: users and devices. Therefore, in RCDA actors have to be described according to their input and output interactive capabilities. These capabilities are mapped to support tasks completion on the interactive system. This mapping takes into account the description of the current task in terms of interactive capabilities. As a result, the interaction adapts to the task performed by the interactive system and the interactive capabilities of users and devices. The process of mapping interaction capabilities to tasks is still an open topic. But previous approaches can be used such as defining roles [4] or defining fitness values [6], among others.

2.2 Adaptations Responsibility

The adaptation of user interaction is responsibility of the system. Are applications and not users who have to adapt and move interaction across multiple devices to help users to fulfil their tasks. They are in charge of mapping interactive capabilities of both users and devices. And, under specific circumstances, users will be able to decide among multiple interaction paths. Depending on how users are able to decide among multiple interaction paths we have set three schemas:

1. Fixed. Users do not decide how the interaction is distributed.
2. Restricted. Users will be able to decide how interaction is distributed in a limited way. They do not have fully control on how to distribute interaction.

3. Free. Users are fully capable of distributing interaction among the interactive system.

Each one of these schemas has it strengths and weaknesses. While the first schema (fixed) is the most restrictive, not allowing users to decide how interaction is distributed, it is also the most simple from the point of view of users. They only have to use the system. They are not aware of when, why and how adaptations are made. In the other hand, the free schema is the most complicated for users. They have to decide when and how the distribution has to be made. Therefore, it is important to support the schema that best fits with the interactive system requirements.

2.3 Runtime Adaptations

RCDA has to dynamically categorize devices and users to change interaction to best-fit users' tasks completion. It is not enough to describe application's scenario statically. Multi-device scenarios tend to change their configuration adding or removing devices/users and consequently changing interaction capabilities. RCDA has to be aware of these circumstances. Therefore, they have to be up to date of the input and output interactive capabilities of the involved actors, adapting interaction in runtime. Therefore, RCDA has to support a flexible mapping between these capabilities and the task user/s is/are performing.

2.4 User Patterns

RCDA has to identify and support users' patterns for tasks execution in cross-device scenarios. Depending on how the task is transferred between devices, the information moves between them or their physical or logical arrangement patterns are classified as serial or parallel. In the serial or sequential pattern the task is transferred from one device to another [4] [3]. One specialization of this pattern is the producer-consumer pattern where the content is produced in one device for been moved to another device to use it [4]. In the parallel pattern many devices are coordinated for simultaneous use [4]. In the performer-informer pattern, one device has the resources to support the creation of content in other device [4] . In the performer-informer-helper pattern users use support devices (helpers) to perform traversal tasks [4]. In the controller-viewer/analyzer pattern users wanted to combine devices functionalities to perform different aspects of a single primary task [4]. Users also perform resource lending (borrowing resources from other devices), related parallel use (all devices are involved in a single task) and unrelated parallel use (devices are involved in different tasks) [3]. Devices can also be physically organized mapping their physical position within the environment, following the spatial mapping pattern [2]. But there could be no spatial mapping when organizing devices [2]. The devices = windows pattern is used when devices are used as windows in the WIMP paradigm [2]. In the device = role pattern each devices is mapped with a single role for the current task [2]. In the device = storage pattern each device shows, for example, an specific document for the task [2].

3 Conclusions and future work

In this paper we introduced the foundations of a new approach to handle with the distribution of interactions in cross-device applications called Responsive Cross-Device Applications (RCDA). The main characteristics for this approach are: go beyond screen size to design adaptations, applications are in charge of adapting interaction in cross-device environments, not users; and make it happens dynamically. Also, depending on how the task is transferred between devices, how the information moves between them or their physical or logical arrangement presented patterns have to be taken into account for the development of RCDA.

As a future work, due to the complexity of RCDA, we will continue with the development of tools for the description and simulation of such a complexes scenarios.

Acknowledgments. This work has been partially supported by the fellowship 2014/10340 from the University of Castilla-La Mancha.

References

1. Chi, P., Li, Y.: Weave: Scripting Cross-Device Wearable Interaction. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 3923-3932. DOI=http://dx.doi.org/10.1145/2702123.2702451 (2015)
2. Hamilton, P., Wigdor, D.J.: Conductor: enabling and understanding cross-device interaction. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 2773-2782. DOI=http://dx.doi.org/10.1145/2556288.2557170 (2014)
3. Jokela, T., Ojala, J. and Olsson, T.: A diary study on combining multiple information devices in everyday activities and tasks. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15, páginas 3903–3912. ACM, New York, NY, USA, ISBN 978-1-4503-3145-6 (2015)
4. Santosa, S., Wigdor, D.: A field study of multi-device workflows in distributed workspaces. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (UbiComp '13). ACM, New York, NY, USA, 63-72. DOI=http://dx.doi.org/10.1145/2493432.2493476 (2013)
5. Schreiner, M., Radle, M., Jetter, H., Reiterer, H.: Connichiwa: A Framework for Cross-Device Web Applications. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15). ACM, New York, NY, USA, 2163-2168. DOI=http://dx.doi.org/10.1145/2702613.2732909 (2015)
6. Yang, J., Wigdor, D.: Panelrama: enabling easy specification of cross-device web applications. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 2783-2792. DOI=http://dx.doi.org/10.1145/2556288.2557199 (2014)