User Interface Distribution Method Based on Pi-Calculus

Sergey Sakulin, Bauman Moscow State Technical University, Moscow, Russia
Alexander Alfimtsev, Bauman Moscow State Technical University, Moscow, Russia
Evgeny Tipsin, Bauman Moscow State Technical University, Moscow, Russia
Dmitry Sokolov, Bauman Moscow State Technical University, Moscow, Russia

https://orcid.org/0000-0002-3805-4499

Vladimir Devyatkov, Bauman Moscow State Technical University, Moscow, Russia
Sergey Sakulin, Bauman Moscow State Technical University, Moscow, Russia

ABSTRACT

The rapid growth of computing devices has led to the emergence of distributed user interfaces. A user interface is called distributed if a user can interact with it using several devices at the same time. Formal methods for designing such interfaces, in particular methods for the distribution of interface elements across multiple devices, are yet to be developed. This is the reason why every time a new application requires a distributed user interface, the latter has to be designed from scratch, rendering the entire venture economically inefficient. In order to minimize costs, unify and automate the development of distributed interfaces, we need to formulate general formal methods for designing distributed interfaces that will be independent from a particular application or device. This article paper proposes a formal distribution method based on the pi-calculus.

KEYWORDS

Distributed User Interface, Distribution Rules, Formal Method, Graphic User Interface, Interface Distribution Method, Pi-Calculus, System Usability Scale, Usability Testing

1. INTRODUCTION

The rapid growth of computing devices has led to the emergence of distributed user interfaces (DUI). A DUI provides for interaction between a user and an interface using several devices simultaneously, which increases application usability (Elmqvist, 2011). Presently, the following DUI implementations exist:

- Synchronization system between a smartphone and a smartwatch, where the latter can perform some of the smartphone functions (Nascimento, Oliveira, & Tam, 2018);
- Device management using a smartphone or a tablet (Stojkoska & Trivodaliev, 2017);
- DUI for generating a shopping list at a supermarket (Ghiani, Manca, & Paternò, 2015);
- DUI for collaborative decision making (Widjaja & Takahashi, 2016).

The said DUI implementations are developed using supplementary software tools (Nebeling & Dey, 2016; Yang & Wигдор, 2014; Chi, Li, & Hartmann, 2016). The tools in turn rely on distribution

DOI: 10.4018/IJDST.2019070101
templates (Tesoriero & Altalhi, 2017) presented in special formats. In particular, the MARIA format (Manca & Paternò, 2011) is designed to describe a distributed abstract interface and its implementation across various devices. The indicated tools and templates do not contain distribution rules. They are used for developing software for particular applications and devices. In the article (Penalver, López-Espín, Gallud, Lazcorreata, & Botella, 2011) devoted to device synchronization in the DUI, the authors introduce the following formal synchronization properties: correctness, consistency, decomposability, portability, simultaneity, and continuity. But distribution rules were not strictly defined.

Every time you design a new DUI, a new dynamic logic for the distribution of interface elements has to be developed as well. This logic may be far from trivial and heavily dependent on developer’s preferences. At this point, it is difficult to introduce the DUI concept to general public as an average user struggles to comprehend the distribution logic behind it and is reluctant to learn new interaction methods (Sakulin, Alfimtsev, Solovyev, & Sokolov, 2018). On the other hand, it implies significant labor effort from developers, considering that the complexity of DUI elements distribution increases exponentially (Gallud, Tesoriero, Vanderdonckt, Lozano, Penichet, & Botella, 2011). For example, to distribute a two-element interface to two devices, you need to consider nine possible scenarios since one element of the DUI can be distributed to either of the two devices or to both.

For the foregoing reasons, developing formal methods for distributing interface elements appears to be an anticipated next step. Section 2.1 contains a formal description of the developed distribution rules. Section 2.3 describes the example adaptation of the pi-calculus to the distribution description. Section 3 contains the results of the usability testing of a distributed interface designed according to the distribution rules from Section 2.1.

2. PROPOSED METHOD

2.1. Distribution Rules for Interface Elements

Each interface $I$ consists of structure blocks. Typical interface structure blocks are (Galitz, 2007):

- Main body $B_1$
- Header $B_2$
- Footer $B_3$
- Left sidebar $B_4$
- Right sidebar $B_5$

The main body is dedicated for user’s common task solving and is located at the center of interface (Garrett, 2010). The header is located horizontally above the main body and left and right sidebars. Each structure block is a set of logical blocks. In turn, each logical block is a set of interface elements that are logically connected. For example, let’s take a look at a simplified interface of an online store (Figure 1). It contains structure blocks $B_1, B_2, B_3$, logical blocks $L_1, L_2$, and elements $E_1, ..., E_{18}$.

The distribution of this interface is shown in Figure 2. Initially, the interface $I$ is displayed on a device $D_1$. Its distributed version $I'$ is located on a main device $D' = D_1$ and an auxiliary device $D_2$. The devices are colored yellow, structure blocks are green, logical blocks are red, and interface elements are gray.

The result of such distribution to the laptop $D_1$ and the smartphone $D_2$, as seen by an end user, is shown in Figure 3.
Towards High Performance Text Mining: A TextRank-based Method for Automatic Text Summarization
[www.igi-global.com/article/towards-high-performance-text-mining/153970?camid=4v1a](www.igi-global.com/article/towards-high-performance-text-mining/153970?camid=4v1a)