Pichiliani, Mauro C.; Dewan, Prasun; Hirata, Celso M. (2018): Lacomo: A Layer to Develop Collaborative Mobile Applications. In: Proceedings of 16th European Conference on Computer-Supported Cooperative Work - Demos and Posters, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/ecscw2018_p4

Lacomo: A Layer to Develop Collaborative Mobile Applications

Mauro Carlos Pichiliani, Prasun Dewan, Celso Massaki Hirata
IBM Research, Univ. North Carolina-Chapel Hill, ITA
mpichi@br.ibm.com, dewan@cs.unc.edu, hirata@ita.br

Abstract. Nowadays, there is little support for developers to transform single user applications to collaborative ones in the mobile domain. We present Lacomo, a new software layer to build collaborative mobile applications with accessibility, screen sharing, and application rewriting technologies that reduce costs to prototype collaboration features, thereby increasing the range of supported applications without deep application knowledge. We comparing it to an *ad hoc* approach. Users using Lacomo performed a testing task faster, with less effort and errors at a higher completion time.

Introduction

Mobile applications are a major force behind the explosive growth of mobile devices. While they greatly extend the functionality of those devices, they also open opportunities to enable collaboration, especially with applications familiar to users.

Leveraging single-user commercial systems to multi-user collaboration has been a persistent research topic. However, so far, related research academic work and the state of the practice approaches pay little attention to the mobile domain, and the approaches that do so either focus in resources that demand high development effort (Lin et al., 2007) or do not cope with specific characteristics of mobile devices (Picco et al., 2014).

To overcome the above limitations, we propose a Layer to develop collaborative mobile applications (Lacomo). Inspired by operating system's accessibility layers, our approach has the goal to promote collaboration through data sharing in user interface widgets found in existing applications, such as text box and buttons. This is a contrast to traditional approaches to alleviate development through source code modification, component placement, or framework reuse.

Related Work

The main supporting infrastructures, frameworks, components and other approaches are reviewed by (Roussev 2003) and presented in Figure 1.

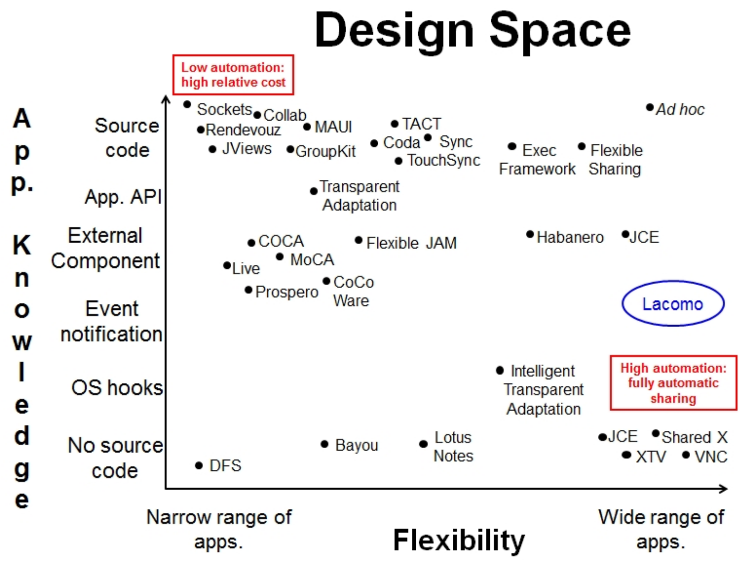

Figure 1. Design space for collaboration support.

The development frameworks, toolkits, and components found on the top left area provide low automation and demand high development relative cost. In contrast, sharing infrastructures located on the lower right provide fully automatic sharing without development effort, but with limited interaction, as to strict WYSIWIS interaction mode. The other approaches in between tend to require high effort development effort and/or constraint flexibility.

The constraints of existing systems limit the range of applications and interaction modes, thus we aim to overcome those limitations and fulfill the requirements further than existing approaches. To achieve this goal we studied the design space to find a suitable place for Lacomo in such a way that it fits most applications (flexibility), with a high level of automation to achieve a considerable amount of code reuse and be extensible to custom modifications with less development effort than previous approaches.

The challenge of designing a single mechanism to address general requirements without the constraints leads us to explore innovative technologies and combine old ones. Specifically, we were interested in providing such a mechanism that automates UI data sharing in relaxed WYSIWIS mode without the full knowledge of objects semantics and application's internal details.

After the study of previous approaches in Figure 1, we detected past systems do not explore event notification and resources provided by the OS to develop accessibility services for mobile platforms. The use of these resources combined with screen sharing, UI automation testing, and application rewriting technologies moved our attention to the area where a wide range of applications can be modified without components, APIs, source code or direct OS hooks. This is advancement over the JCE, XTV, VNC and other approaches that do not require the source code, since the resources we combine provide contextualized UI widget data and event information.

Lacomo Design

Lacomo is designed to be built on top of an API that capture user events, such as the accessibility API, thus any existing application that has widgets compatible with accessibility interfaces can benefits from its features.

The services developed with Lacomo can access events and data associated with elements of the UI interface without application source code in the same way as any accessibility service. Additionally, since Lacomo services can access the System log, every communication from the application to the OS and vice-versa is available to the Lacomo layer. The hierarchy of objects that corresponds to UI is available to the developer and every pixel shown is reachable due to screen sharing technology. These features achieve a high level of code reuse since they are available at run time and do not require the source code

User Study

The goal of the user study is to compare the Lacomo implementation with the *ad hoc* to provides non-WYSIWIS collaboration in an existing mobile application. Formally, this experiment aims to collect quantitative and qualitative evidence to verify the following hypothesis: *Does the use of Lacomo require less effort to modify an existing mobile application than the ad hoc technique?*
20 participants (2 female, 18 male), with age ranging from 22 to 53, participated in the experiments. The effort metrics evaluated were Time, LOC (Lines of Code), LOC divided by time, Calories, Mouse movements and Save events. Figure 2 shows a radar chart that compares the normalized mean values for the five effort variables and the time spent during the study.

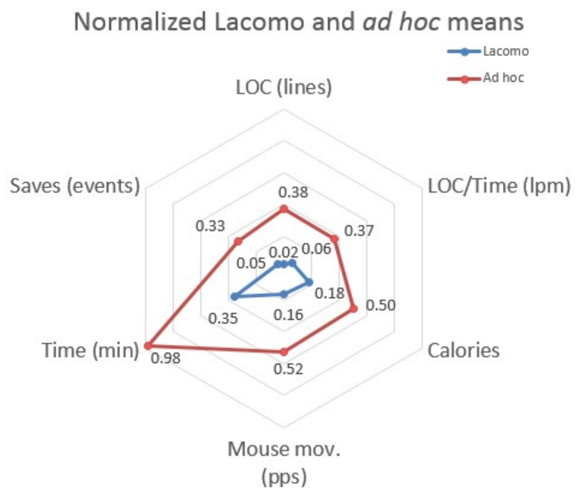

Figure 2. Normalized values of the effort metric.

The effort required to change a mobile app to support collaborative requirements is, on average, 27.7 minutes, 11.3 lines of code, 0.51 lines per minute, 97.1 calories, 224,841.1 pixels traveled by the mouse, and 6.7 save events. The effort to use an event notification approach compared to *ad hoc* implementation required, on average, 64% less time, 95% less lines of code, 78% less lines per minute, 64% less calories, 69% less pixels traveled by the mouse, and 85% less save events.

In general, participants commented that the Lacomo approach was interesting. P8: "*It was interesting to see a new way of development*". P16: "*I liked, and it [Lacomo] was very interesting. It would open other [development] possibilities*". When asked about the difficulty they faced, participants of the ad hoc group reported problems to understand the application code and to capture events. P17: "*[I could not] make the client event appear on the next tablet*". P13: "*It was hard for me to understand the large [existing] code and find the elements [needed]*"

References

Lin, K. Chen, D., Dromey, R., Xia, S., Sun, C. API Design Recommendations for Facilitating Conversion of Single-user Applications into Collaborative Applications. In Proc. 3rd CollaborateCom (2007), 309-317.

Picco, G.P., Julien, C., Murphy, A.L., Musolesi, M., Roman, G. Software Engineering for Mobility: Reflecting on the Past, Peering into the Future. In Proc FOSE 2004 (2014) 13-28.

Roussev, V. Flexible Sharing of Distributed Objects Based on Programming Patters. PhD thesis at University of North Carolina, USA (2003).