Mobile-IDM: A Design Method for Modeling the New Interaction Style of Mobile Applications

Giovanni Coppola, Alessandro Fiore(✉), Luca Mainetti, and Andrea Pandurino

Department of Innovation Engineering, University of Salento,
Via Monteroni 73100, Lecce, Italy
{giovanni.coppola,alessandro.fiore,luca.mainetti,
andrea.pandurino}@unisalento.it

Abstract. Mobile applications (apps) offer designers the opportunity to experiment with novel interaction grammars (e.g., gestures, context-aware events), whose implications for conceptual modeling still need to be fully understood. The research panorama only proposes a few design methods for apps, which are mainly released as extensions of existing ones. This, in addition to the short lifecycle that characterizes apps, leads to the risk of inappropriate modeling techniques being adopted. To bridge this gap, we propose a new design method, named Mobile-IDM, to model the interaction between the user and the app from a logical point of view. As it is based upon IDM and Rich-IDM, from which it inherits its design semantics, Mobile-IDM exploits the dialog metaphor to facilitate the establishment of a common ground between designers and web engineers to obtain good usability of the interaction. We demonstrate through a case study the simplicity and other advantages of our approach.

Keywords: User experience design · Mobile application · Design methodologies · IDM · Rich-IDM

1 Introduction

The statistics related to the uptake and the use of mobile devices, such as tablets and smartphones, are staggering. Just consider the fact that in the world 1,75 billion people use a smartphone in 2014 [1] and, in many European countries, it is not unusual to find people who have more than one device.

Technological development has also led manufacturers to produce devices with more powerful processors, more capacious memories, and more sophisticated hardware innovations (such as GPS, touchscreens, accelerometers, NFC, and other types of sensors). By combining these developments with the increasing diffusion of high-speed wireless Internet connections, it is possible to use mobile devices to perform a series of daily life tasks.

To satisfy the needs of the users, the number of applications present in online stores has exploded. At the end of 2014, the Google Play Store and Apple App Store contained about 1.43 million and 1.21 million applications, respectively. Actually, many of these have very similar features, allowing the user to perform the same

P. Cimiano et al. (Eds.): ICWE 2015, LNCS 9114, pp. 101–115, 2015.
DOI: 10.1007/978-3-319-19890-3_8

activities in a different way. Of course, the user will choose a specific application instead of another one according to their satisfaction in interacting with it.

For these reasons, it is necessary to rethink the user interface (UI) design methods for mobile applications in order to improve the quality of user interaction and, in general, the user experience. An important aspect that needs to be considered is the type of device on which the application will be used, as there are devices with different screen sizes and with different operating systems (Android, iOS, Windows Phone). These systems have different user interface elements and they already provide their own guidelines for interface modeling (for example, Apple does not permit the application's distribution on the App Store if these guidelines are not followed).

In this paper, in order to cater for some new distinctive characteristics enabled by mobile applications, we propose a design method for mobile applications called Mobile-IDM. Mobile-IDM is an innovative extension of Rich-IDM (Rich Interactive Dialogue Model) [2][3], which is traditionally used for modeling Rich Internet Applications (RIAs). Mobile-IDM integrates the theoretical constructs of dialogue theory (defined in IDM [4]), enabling the fundamental user interaction requirements, with the interaction and navigation grammar of the mobile device, to be captured. As a result, our approach aims to serve as a simple conceptual common ground between advanced design features and user interaction concerns. The features of Mobile-IDM, which are based on a few, simple conceptual primitives, allow us to hypothesize some key advantages: they are expressive enough to efficiently model the interaction between the user and the mobile application and they are structured enough to be directly connected to the user experience requirements as captured in IDM. In the last part of this work, we apply Mobile-IDM to a real case study in order to better describe its characteristics and to demonstrate the advantages of our approach.

The paper is organized as follows. The related work on web and mobile application design is summarized in Section 2. Section 3 presents the open issues that exist in the area of mobile application modeling approaches. A brief introduction on the IDM framework is reported in Section 4. Section 5 presents the Mobile-IDM approach, explaining its notation and semantics. Section 6 illustrates an application of Mobile-IDM (through a real case study) in order to address the issues highlighted in Section 3. Finally, in Section 7 the conclusions summarize our key messages and outline future research directions.

2 Related Work

In this section, we provide a general survey of the related work about web or mobile application design. In the literature, we can find several approaches for designing browser-based applications.

Hera [5] is a design methodology composed of several steps that define the many aspects of a system using models with graphical notations. This approach is supported by the Hera Presentation Generator, a tool capable of generating webpages from Hera models.

In [6], the authors propose a development framework for web applications on heterogeneous devices. The framework is based on a semiautomatic process capable of customizing the application for specific target devices and transforming HTML pages into generic application artifacts.

The Web Modeling Language (WebML) [7] is a notation for modeling web applications at the conceptual level. It enables the high-level description of a website under distinct orthogonal dimensions: its data content (structural model), the pages that compose it (composition model), the topology of links between pages (navigation model), the layout and graphic requirements for page rendering (presentation model), and the customization features for one-to-one content delivery (personalization model).

The Interaction Flow Modeling Language (IFML) [8], an extension of WebML, is a modeling language dedicated to the definition of "User Interaction" dynamics. The IFML language is abstract and independent from the implementation technology. As a result, IFML supports the platform-independent description of graphical user interfaces for applications accessed or deployed on such systems as desktop computers, laptop computers, PDAs, mobile phones, and tablets. The focus of the description is on the structure and behavior of the application as perceived by the end user.

TERESA [9] is a tool that allows the development and UI generation for multi-device interfaces starting from tasks defined by CTT (Concur Task Tree) notation. TERESA is able to support UI generation for a wide set of implementation languages: XHTML MP, VoiceXML, X+V, SVG, Xlet, and the gesture library for MS. This means that designers and developers have a consistent environment that allows them to build applications for a variety of platforms, which support various modalities.

RUX [10] designs the UI of RIA applications through four levels of definitions. The concepts and tasks level contains the data and business logic and can be modeled using a web design methodology, such as in WebML. The RUX-model strategy is fully compliant with many web design methodologies such as WebML or UWE (UML-based Web Engineering [11]).

UWE uses a specific UML profile and allows the model-driven development of web applications. The UWE design process has a requirement model, a content model, a navigation model, and a presentation model, which provides an abstract view of the Web UI.

OOHDM [12] proposes a design process where the interface and its behavior are explicitly analyzed and examined. The implementation step of the OOHDM design process is platform-dependent and maps the conceptual design on the specific RIA technology. In this step, the static aspect of the interface is modeled using Abstract Data Views (ADV) [13] while ADV-charts [14] are used to define the status and the behavior of each interface element. The OOHDM approach describes the application pages by defining their views and their charts to model the changes. The ADV approach has also been used within the process of Web Model Refactoring [15] towards RIA. This approach aims at transforming standard web interface elements into others, adding new interaction facilities that capture RIA features.

These methods, even if not properly designed for mobile applications, may be used in this field thanks to their platform independence. Furthermore, they offer useful guidelines about user interaction modeling. Nevertheless, a few methods explicitly intended for mobile applications can be found in the literature.

In [16], the authors propose a maieutic process for modeling mobile applications by taking into account the user's point of view. This approach starts by detecting the needs of the users and ends by transferring them into a system of functions and applications without limiting the satisfaction of superficial requirements.

In [17], the use of market elements is recognized as a leading factor in the success of a mobile software product. Thus, the authors recommend the use of the market-based New Product Development (NPD) for improving current mobile development.

In [18] an agile approach to mobile application development called Mobile-D is reported. Mobile-D is based on development practices borrowed from XP (eXtreme Programming), enjoys the scalability of methods inspired by the Crystal family of methodologies, and provides life-cycle coverage as defined by the RUP (Rational Unified Process).

Finally, MobileIFML [19] proposes an extension of IFML that is tailored to mobile applications. MobileIFML comprises the development of automatic code generators for cross-platform mobile applications based on HTML5, CSS, and JavaScript.

3 Open Issues

Considering the approaches discussed above, it is important to notice that many of the examined methodologies do not cover all of the specific aspects of mobile applications. For example, RUX does not consider the semantic and communicative value of the elements that compose the UI. Thus, it does not model the information flow that leads the user during navigation. In Hera the flow is modeled from a more conceptual perspective. None of the analyzed methods directly refers to mobile UIs, although we can find a number of important points for consideration. For instance, Hera and WebML provide a way to represent contextual information and how to use it. Furthermore, WebML suggests using several views for modeling the UI, one for each platform target.

In addition, the considered research on mobile applications does not focus on the user experience but rather on the design process in general. In particular, MobileIFML faces the problem from a technological and implementation perspective, providing a tool for automatic code generation. Our approach, instead, allows user experience to be designed for mobile applications starting from a conceptual level, which supports requirements analysis.

Most of the time clients require mobile applications to be developed as swiftly as possible and for multiple platforms and devices. From the technological perspective, the increasingly frequent use of cross-platform frameworks (e.g., PhoneGap, Titanium, Xamarin) testifies to this trend. Therefore, conceptual design becomes essential in the development of mobile applications in order to avoid errors in modeling functionalities and user interaction, even at the expense of technological aspects. Having a methodology that allows designers to model these aspects is a key factor in the success of a mobile application.

4 Introducing the IDM Methodology Framework

The IDM framework (Fig. 1) uses the theory of dialogue as the basis for describing human–computer interaction, considering every element of the interface as a dialogue fragment. The interface's elements are built using several dialogue types and techniques (such as form filling, menu selection, icons, etc.). Thus, suitable modeling is based on the collaboration between domain and communication experts, and user experience experts and engineers.

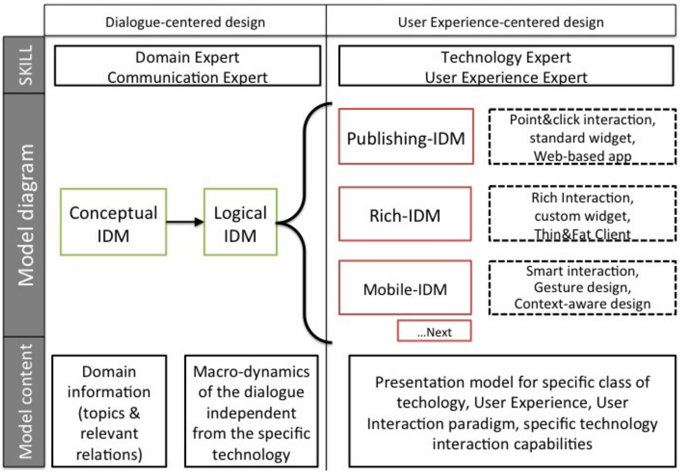


Fig. 1. IDM Methodology Framework

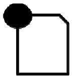
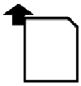
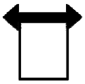
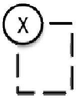
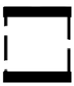
The first phase of the design is to describe the application at the conceptual level, focusing on the dynamics of the dialogue: the IDM primitives [4] can be used to catch the essential interactive and navigation features of information-intensive applications. These primitives are organized in two main design layers: Conceptual IDM (C-IDM) and Logical IDM (L-IDM). C-IDM is used to describe the "conceptual schema" of the application, defining the topics of the dialogue and the relations between its elements. In other words, it is used to shape the deep dialogic structure of the interaction [20]. Starting from the C-IDM design, the L-IDM schema models the decisions that are typically dependent on a specific fruition channel through which the application may be conveyed.

The second phase of the framework, instead, focuses on the user experience design, which is based on the specific class of technology. In particular, Publishing-IDM (P-IDM) considers several design aspects of the webpages, such as the layout or the frame position inside the page, while Rich-IDM is used for modeling RIAs.

4.1 Adapting Rich-IDM to Mobile Applications

Rich-IDM [2] [3] is a methodology for modeling the user experience of RIAs. It is currently being used in several research and industrial projects as it provides a description of the complex interaction offered to users in RIAs, including non-trivial relationships, behaviors, and views. Rich-IDM provides several primitives in order to model all the specific aspects of RIAs. In particular, the UX Core Element and the Context View Element allow the user's attention to be focused on the main elements of the dialogue, a fundamental requirement given the large amount of information contained in rich Internet pages. Table 1 describes the notation and the semantics of Rich-IDM primitives.

Table 1. The Rich-IDM design primitives

Name	*Notation*	*Design semantics*
Content RIA-Page Element		A coherent, atomic fragment of a RIA page, which displays a *content* unit
Introductory RIA-Page Element		A fragment of a RIA page which displays mechanisms to enable *access* to multiple instances of a topic
Transition RIA-Page Element		Allows users to follow the semantic relation of two topics
User Experience Core		A connected composition of page elements, which communicates the semantic nucleus of what is offered to the user at a given moment
RIA-Handle		An interaction affordance, which enables users to move within two or more page elements of the same user experience core
Context View		A set of user experience cores which maintain navigational context, orientation, and fluid transition between the cores
Default Element		Indicates the default RIA-Page Elements

The main Rich-IDM features are: i) expressiveness to capture interactive features at a high level of abstraction; ii) semi-formality to facilitate the establishment of a common ground between designers and software engineers; and iii) traceability of the design to important usability requirements. For these reasons, this method is well suited to modeling the user experience in the case of mobile applications, although it is necessary to use in a different way some primitives adapting them to the new contexts. Unlike RIAs, the screen size that is available in mobile devices is usually very limited for the purpose to show information to the user. Thus, in order to define and to organize the informational content, it is fundamental to redefine the User Experience Core usage. In fact in RIA, the UX Core element is used to indicate the objects containing the dialogue focus. It contains the dialogue elements that are the reasons for which the page has been created. Instead, in a mobile application, the UX Core designs the fundamental elements for satisfying the requirements of the page. Furthermore, the RIA-Handle primitive assumes an even more important role. In RIAs, users have the possibility of interacting with the page elements in different ways (e.g. drag & drop) using only the mouse in order to perform the actions. In this case, the RIA-Handles describe how the user can activate the transition to the target using standard input methods (e.g. mouse, touchpad). Instead in a mobile app the user can use many gestures (see fig. 5) to interact with the content but he/she can perform only a limited numbers of actions due to the screen sizes (e.g. drag & drop does not exist). Thus, the role of the RIA-Handle is limited in the mobile application but we have to consider the important function of gestures to optimizing the ergonomics of content navigation. In detail, it is important not only the use of the gestures but also how the gestures are used; e.g. it is our opinion that many swipes to see all the page content does not help the user experience.

Finally, also the Context View primitive is useful to model user interfaces of mobile application. In RIAs, it allows gathering related UX Cores with a same “look&feel” defining a specific navigational context; thus, it provides to the user the concept of section of RIA site. The mobile application characteristics suggest the use of a single Context View because the mobile apps are vertical application focused on specific aspect and, thus, the change of context is not needed. Moreover, the change of “look&feel” may confuse the user. Consequently, in the case of reverse engineering, the presence of more Context View could indicate an incorrect modeling of the application.

However, mobile applications have specific features that the Rich-IDM method is not able to model correctly.

5 Mobile-IDM

In this section, we provide readers with an introduction to Mobile-IDM, presenting its notation and primitives in order to overcome the lacks of Rich-IDM in the context of mobile application design. Starting from a common L-IDM information design, several M-IDM models of the same application, one for each combination device/orientation, can be obtained with Mobile-IDM. This can lead to a large number of models, although they are more readable than a single model in which all possible combinations are specified as variants. Table 2 illustrates the Mobile-IDM notation and its design semantics. Then, it follows a brief description of the distinctive aspects of mobile applications and how they can be modeled using Mobile-IDM.

5.1 Context-Aware Identifier

Modern mobile devices are able to discover and take advantage of contextual information such as user location, time, user activity, and preferences. These types of information (often obtained by the sensors on the device) are used by mobile applications to automatically adapt their behavior according to the context. Thus, it is possible to improve the user experience by providing users with what they probably need in a specific place, at a specific time, and for a specific task. However, in case of mobile applications, the question is: “How can the designer model the elements in the mobile application that are related to the context?”

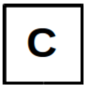
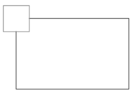
In order to resolve this need, Mobile-IDM introduces the *Context-aware Identifier*. It models the dialogue elements that contain context-aware information. It may be associated with a single RIA Page Element or to a User Experience Core. Additional specifications can be attached to Context-Aware Element through textual annotations. For example, the designer can describe how the marked element adapts its content or its appearance (see Fig. 2). Thus, the Context-aware Identifier makes explicit the context-aware behavior of information elements in accordance to the goal of providing an easy to use tool.

5.2 Notification Identifier

A push notification allows an application to notify the user about updated contents or events, thus creating a new interaction flow with the user. The notifications become a point of access to the application, directly showing the updated content and making the interaction suddenly change.

It is needed to provide designers a solution for indicating the elements that are communicated to the users through a notification. How could we do this?

Table 2. The Mobile-IDM notation and design semantics

Name	*Notation*	*Design semantics*
Context-aware Identifier		Model the elements of the mobile application that contain context-aware information
Notification Identifier		Model the element that, after a notification, can become the first cue of the dialogue
Interaction Core Container		Model the user interaction with multiple RIA-Page Elements, describing navigation and presentation according to a specific strategy
Landmark Identifier		Indicate the RIA-Page elements that should be accessible from anywhere

Mobile-IDM offers the *Notification Identifier* to specify the above elements. Formally, this primitive says that the RIA-Page Element gets displayed when the user opens a notification. Again, the designer can add a textual annotation in order to better describe the interaction (Fig. 3). Considering the IDM dialogue metaphor, we can imagine the notification as the beginning of a dialogue made by the application itself rather than by the user.

5.3 Landmark Identifier

The limited screen size of mobile devices forces the user to navigate through several screens before reaching the desired information. Thus, in order to improve the user experience, is essential that a mobile design method offers a primitive able to model a direct access to specific application elements.

In web applications, the landmarks are often used allowing a quick access to single information elements (instances) such as the homepage, contacts, credits etc. Mobile-IDM extends the semantics of landmarks, providing designers with the *Landmark Identifier*, which models a dynamic link to a specific information class – an entity, or a topic according to the dialogic metaphor – or an application feature such as highlights, point-of-interest etc. Thus, the linked information could be not simple static elements (such as an individual topic or a RIA Element) but dynamic ones, typically related to the current context of application such as its spatial position and time.

Using the Landmark Identifier, the designer can specify the linked contents by putting them into a container marked with the label "L" (see Fig. 4). Formally, the landmarks are modeled with the Introductory RIA-Page Elements that should be accessible from anywhere. The Landmark Identifier primitive could be combined with the Context-aware Identifier and the Notification Identifier to model the context-aware and dynamic behavior of the information flow.

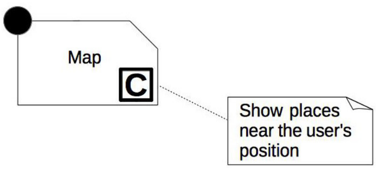


Fig. 2. Example of Context-Aware Identifier

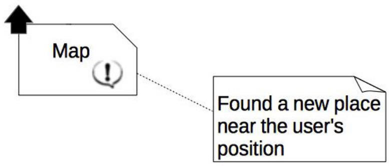


Fig. 3. Example of Notification Identifier

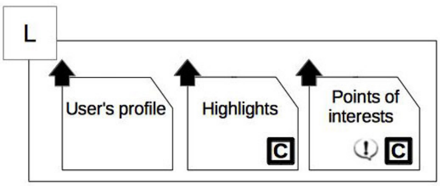


Fig. 4. Example of a Landmark Identifier

5.4 Interaction Core Container

In a mobile application it is needed to establish the best navigation strategies – according to the type and the quantity of contents – to ensure a good user experience. For example, in some cases it is useful to display the contents along a predefined order (Guided Tour strategy), in others it is convenient to navigate in a free manner (All-to-all strategy), and in some circumstances it is better to adopt a mixed strategy or to show all contents at the same time, without any transition between them (Simultaneous Presentation). How can the designer be encouraged to explicitly model these aspects?

Mobile-IDM meets this need providing designers with the *Interaction Core Container*. It is a container of RIA-Page Elements that belong to the same User Experience Core. In the Interaction Core Container the designer can specify one of the four Navigation Strategies defined in IDM (Guided Tour, All-to-all, Mixed, Simultaneous presentation) and an interaction mode from a list of basic gestures (see Fig. 5). This list can be extended with other elements to fulfill the designer's needs. In order to make the modeling more stringent, it can be added a number to define how many actions are needed to navigate all elements within the Interaction Core. For example, a “Swipe/Flick icon” associated with a number “2” defines that the swipe gesture is limited to a length equal two times the width of the screen. The absence of this number models an interaction without a predefined number of gestures.

Considering the IDM semantics, the Interaction Core Container allows designers to describe how the dialogue takes place, according to a specific strategy.

Fig. 5. Interaction modes

6 Case Study: Visit Trentino Tourist Guide

To demonstrate the effectiveness of our design method, we modeled an existing mobile application – Visit Trentino Tourist Guide. This app aims to help tourists organize a holiday in the Trentino region of Italy, providing information about places, sport centers, reception facilities, and other points of interest or services.

On the basis of our consideration about the context view primitives (explained in the 4.1 paragraph), we assume that there is only one context view; thus, the following diagrams do not report the context view in order to improve its readability.

For a lack of space, here we only analyze the topic "Discover point of interest," which allows the exploration and discovery of attractions thanks to the Introductory Acts shown in Fig. 6. These elements provide access to various categories of points of interest, giving a strong semantic value to the dialogue.

The first part of this case study is based on a reverse engineering of the application, which is actually only available for smartphones in portrait mode. In this way we demonstrate the descriptive capabilities of Mobile-IDM. The second part of the case study focuses on the outcome of a forward engineering process and allows the effectiveness of Mobile-IDM to be demonstrated.

Fig. 7 shows the Mobile-IDM model related to the smartphone with portrait orientation scenario. It is mapped with several screenshots of the application in order to demonstrate how the UIs are graphed with Mobile-IDM. For example, let us suppose that the user wants to search for the most interesting tourist places in the Trentino region.

The interaction starts from the main screen of the application by selecting the Introductory RIA-Page Element. Next, the user browses through a series of other pages that filter the results ("Tourist Sector" and "Types of touristic places") until the application shows a list of possible points of interest. The user can select (with a tap gesture) a particular point of interest to view its detail page. This page is modeled with four RIA-Page Content Elements grouped into an Interaction Core, which also defines the navigation strategy and the interaction mode. In particular, the number "2", which is associated with the notation, defines that the scroll is limited to a length equal to two times the length of the screen.

Thus, we have demonstrated that every aspect related to user interaction can be correctly represented with Mobile-IDM. Moreover, the resulting model is easy to read for designers that do not have specific skills in web engineering.

During the forward engineering phase, we supposed a landscape version of the application. A possible modeling is shown in Fig. 8. Thanks to the greater width of the screen, the designer can specify more Introductory RIA-Page Elements, which are grouped into Interaction Core labels. Obviously the height of the screen is smaller compared with portrait mode, so the designer must specify a larger number of vertical scrolls to display all the contents. To model this aspect we added the number "3" near to the icon of the scroll gesture.

Furthermore, we added a Notification Element and a Context-Aware Element to design the application behavior in the presence of context information. When the user displays the map in the information page of a point of interest, the application shows both the point's position and the user's. Moreover, the system notifies the user of the presence of other points of interest in the neighborhood of the user. A single tap on the corresponding notification allows the user to directly open the information page of that point. Finally, Fig. 9 shows a possible model for tablet devices. In this case, the larger size of the screen allows more contents to be displayed simultaneously. When the user selects a desired point of interest, the information pages of other points of interest in the same category can be displayed and browsed with a swipe gesture.

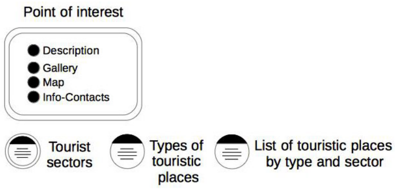


Fig. 6. "Discover Point of Interest" topic: L-IDM model

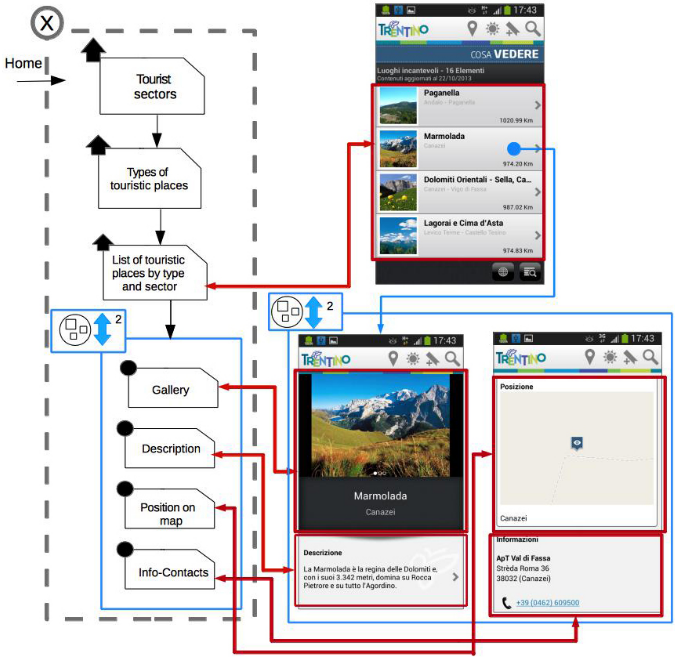


Fig. 7. Mobile-IDM model of Visit Trentino for smartphone (portrait orientation)

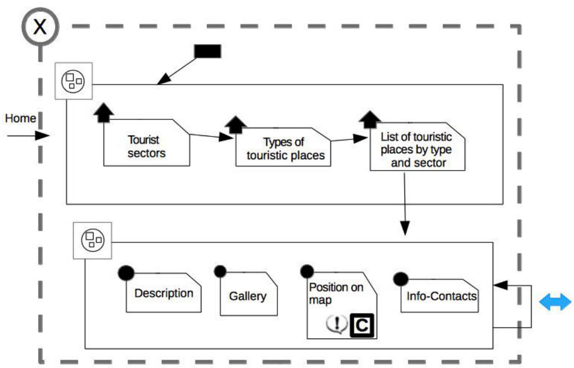


Fig. 8. Mobile-IDM model of Visit Trentino for smartphone (landscape orientation)

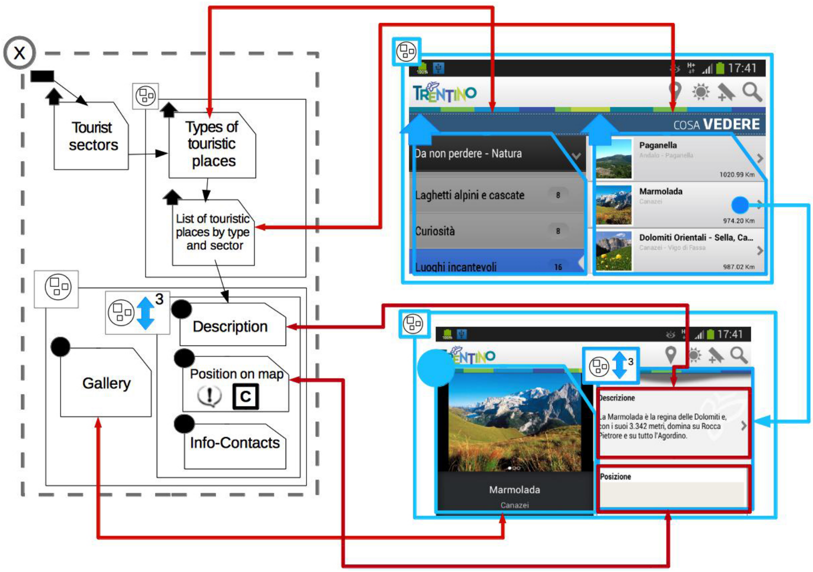


Fig. 9. Mobile-IDM model of Visit Trentino for tablet devices

Mobile-IDM is able to model this interaction simply using a RIA-Handle that comes out and falls into the Interaction Core and the swipe interaction mode. Thus, thanks to Mobile-IDM, we can design the user interaction taking into account screen sizes and device orientation.

7 Conclusions and Future Work

Mobile applications are a real revolution in the field of software engineering, as they have totally changed the interaction paradigm. To balance the trade-off between small screen sizes of mobile devices and the need to ensure good usability, cognitive workload, and efficiency, we have codified a new set of high-level modeling constructs based on the IDM/Rich-IDM approaches. Mobile-IDM primitives provide a common vocabulary between interaction designers (focusing on user requirements) and software engineers (focusing on translating those requirements into feasible and successful solutions).

Furthermore, we have defined some steps toward developing the same application for multiple types of devices, adapting at design time the information architecture and the interaction to the context of the device, with the aim of improving the user experience. Thanks to Mobile-IDM, the interaction between user and application is represented from a logical rather than a technological point of view. This simplifies the communication among designers and engineers as well as aiding the comprehension of stakeholders who do not have computer skills in the development process. Focusing on the design of the user experience and on the semantic modeling of the interface, Mobile-IDM offers a different perspective to mobile application design if compared to other methods currently present in literature. We have shown in a case study the potential of our approach to point out one way to integrate a user experience perspective in the modeling of mobile applications.

Future research directions will be both upstream and downstream. We plan to evolve and validate our requirement engineering reference method to capture other distinctive mobile application needs and to guarantee the artifacts' traceability from the design space to the requirements space. Furthermore, we plan to update our Rich-IDM fast prototyping environment, by adding the Mobile-IDM primitives, to provide designers with means for generating early mobile application mockups.

Acknowledgments. This research was partially funded by the Italian Ministry of Education, Universities and Research (MIUR) under the PON04_a2 DICET-INMOTO (LivingLab on Culture and Technology - Information and Mobility for Tourism) and under the PON02_00563_3470993 VINCENTE (A Virtual collective INtelligenCe ENvironment to develop sustainable Technology Entrepreneurship ecosystems) national research projects.

References

1. Smartphone Users Worldwide Will Total 1.75 Billion in 2014 (n.d.) retrieved April 1, 2015. http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536#sthash.cIJSOYbN.dpuf
2. Pandurino, A., Bolchini, D., Mainetti, L., Paiano, R.: Dialogue-based Modeling of Rich Internet Applications: The Rich-IDM Approach. Int. Journal of Web Information Systems **8**(2), 157–180 (2012)
3. Paiano, R., Pandurino, A., Mainetti, L.: Rich-IDM: transforming the user experience of legacy applications. In: 2011 13th IEEE International Symposium on Web Systems Evolution (WSE), pp. 81–85. IEEE Press, New York (2011)
4. Bolchini, D., Paolini, P.: Interactive Dialogue Model: A Design Technique for Multi-Channel Applications. IEEE Trans. Multimedia **8**(3), 529–541 (2006)
5. Frasincar, F., Houben, G.-J., Barna, P.: Hypermedia Presentation Generation in Hera. Inform. Syst. **35**(1), 23–55 (2010)
6. Banavar, G., Bergman, L., Cardone, R., Chevalier, V., Gaeremynck, Y., Giraud, F., Halverson, C., Hirose, S., Hori, M., Kitayama, F., Kondoh, G., Kundu, A., Ono, K., Schade, A., Soroker, D., Winz, K.: An Authoring Technology for Multidevice Web Applications. IEEE Pervasive Comput. **3**(3), 83–93 (2004)
7. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): A Modeling Language for Designing Web Sites. Comput. Netw. **33**(1–6), 137–157 (2000)
8. Interaction Flow Modeling Language (IFML) Beta1. Object Management Group, Inc., Needham (2013)
9. Paterno, F., Santoro, C., Mantyjarvi, J., Mori, G., Sansone, S.: Authoring Pervasive Multimodal User Interfaces. Int. J. Web. Eng. Tech. **4**(2), 235–261 (2008)
10. Linaje, M., Preciado, J.C., Sánchez-Figueroa, F.: A method for model based design of rich internet application interactive user interfaces. In: Baresi, L., Fraternali, P., Houben, G.-J. (eds.) ICWE 2007. LNCS, vol. 4607, pp. 226–241. Springer, Heidelberg (2007)
11. Preciado, J.C., Linaje, M., Morales-Chaparro, R., Sánchez-Figueroa, F., Zhang, G., Kroiβ, C., Koch, N.: Designing rich internet applications combining UWE and RUX-method. In: Eighth International Conference on Web Engineering, 2008. ICWE 2008, pp. 148–154. IEEE Press, New York (2008)

12. Urbieta, M., Rossi, G., Ginzburg, J., Schwabe, D.: Designing the interface of rich internet applications. In: Latin American Web Conference, 2007. LA-WEB 2007, pp. 144–153. IEEE Press, New York (2007)
13. Cowan, D.D., Lucena, C.J.P.: Abstract Data Views: An Interface Specification Concept to Enhance Design for Reuse. IEEE Trans. Softw. Eng. **21**(3), 229–243 (1995)
14. Carneiro, L.M.F., Cowan, D.D., Lucena, C.J.P.: ADVcharts: A Visual Formalism for Interactive Systems. ACM SIGCHI Bull. **26**(2), 74–77 (1994)
15. Rossi, G., Urbieta, M., Ginzburg, J., Distante, D., Garrido, A.: Refactoring to rich internet applications. A model-driven approach. In: Eighth International Conference on Web Engineering, 2008. ICWE 2008, pp. 1–12. IEEE Press, New York (2008)
16. Gerstheimer, O., Lupp, C.: Needs Versus Technology—The Challenge to Design Third-generation Mobile Applications. J. Bus. Res. **57**(12), 1409–1415 (2004)
17. Vainio, A.M., Tuunanen, T., Abrahamsson, P.: Developing software products for mobile markets: need for rethinking development models and practice. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS 2005, p. 189b. IEEE Press, New York (2005)
18. Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jäälinoja, J., Korkala, M., Koskela, J., Kyllönen, P., Salo, O.: Mobile-D: an agile approach for mobile application development. In: OOPSLA 2004 Companion to the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications, pp. 174–175. ACM, New York (2004)
19. Brambilla, M., Mauri, A., Umuhoza, E.: Extending the interaction flow modeling language (IFML) for model driven development of mobile applications front end. In: Awan, I., Younas, M., Franch, X., Quer, C. (eds.) MobiWIS 2014. LNCS, vol. 8640, pp. 176–191. Springer, Heidelberg (2014)
20. Pandurino, A., Bolchini, D., Mainetti, L., Paiano, R.: Streamlining complexity: conceptual page re-modeling for rich internet applications. In: Harth, A., Koch, N. (eds.) ICWE 2011. LNCS, vol. 7059, pp. 289–301. Springer, Heidelberg (2012)