



## Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering



José Matías Rivero<sup>a,b,\*</sup>, Julián Grigera<sup>a</sup>, Gustavo Rossi<sup>a,b</sup>, Esteban Robles Luna<sup>a,c</sup>, Francisco Montero<sup>d</sup>, Martin Gaedke<sup>e</sup>

<sup>a</sup> LIFIA, Facultad de Informática, UNLP, La Plata, Argentina

<sup>b</sup> Research institute, Conicet, Argentina

<sup>c</sup> Research institute, CIC, Buenos Aires, Argentina

<sup>d</sup> LoUISE Research Group, UCLM, Albacete, Spain

<sup>e</sup> Chemnitz University of Technology, Germany

### ARTICLE INFO

#### Article history:

Received 12 December 2012

Received in revised form 21 January 2014

Accepted 22 January 2014

Available online 29 January 2014

#### Keywords:

Mockups

User-Interface

Agile

Web Engineering

MDD

### ABSTRACT

**Context:** Agile software development approaches are currently becoming the industry standard for Web Application development. On the other hand, Model-Driven Web Engineering (MDWE) methodologies are known to improve productivity when building this kind of applications. However, current MDWE methodologies tend to ignore important aspects of Web Applications development supported by agile processes, such as constant customer feedback or early design of user interfaces.

**Objective:** In this paper we analyze the difficulties of supporting agile features in MDWE methodologies. Then, we propose an approach that eases the incorporation of well-known agile practices to MDWE.

**Method:** We propose using User Interface prototypes (usually known as *mockups*) as a way to start the modeling process in the context of a mixed agile-MDWE process. To assist this process, we defined a lightweight metamodel that allows modeling features over mockups, interacting with end-users and generating MDWE models. Then, we conducted a statistical evaluation of both approaches (traditional vs. mockup-based modeling).

**Results:** First we comment on how agile features can be added to MDWE processes using mockups. Then, we show by means of a quantitative study that the proposed approach is faster, less error-prone and still as complete as traditional MDWE processes.

**Conclusion:** The use of mockups to guide the MDWE process helps in the reduction of the development cycle as well as in the incorporation of agile practices in the model-driven workflow. Complete MDWE models can be built and generated by using lightweight modeling over User Interface mockups, and this process suggests being more efficient, in terms of errors and effort, than traditional modeling in MDWE.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

In the last years, Model-Driven Web Engineering (MDWE) approaches like WebML [1], UWE [2] or OOHDM [3] have become mature solutions for developing Web Applications. These methodologies apply Model-Driven Development (MDD) concepts to capture high-level Web Applications concepts into models and use these models to derive applications automatically. The classic MDWE development process consists of three steps [4]: (1)

building a domain model, (2) defining a hypertext model and (3) defining the application's look and feel. The result of the process is a set of models that can generate the final Web Application through code generation.

While standard MDWE processes improve productivity, they tend to leave User Interface aspects to the end of the development cycle [5]. As a consequence, customers only see their Web Application in action after a complete iteration and, at this point, they may encounter new requirements which may entail a full application rework. Changes in the UI may not only affect the application's presentation, interaction and usability aspects, but also the business logic [6,7], and in the context of MDWE, this may entail potential changes in the models at all three levels. In addition, while using high-level languages facilitates the requirements transformation to working software, the hard dependencies between each modeling step usually slows down the process.

\* Corresponding author at: LIFIA, Facultad de Informática, UNLP, Calle 50 y 120, La Plata, Buenos Aires, Argentina. Tel.: +54 92215082703.

E-mail addresses: [mrivero@lifia.info.unlp.edu.ar](mailto:mrivero@lifia.info.unlp.edu.ar) (J.M. Rivero), [julian.grigera@lifia.info.unlp.edu.ar](mailto:julian.grigera@lifia.info.unlp.edu.ar) (J. Grigera), [gustavo@lifia.info.unlp.edu.ar](mailto:gustavo@lifia.info.unlp.edu.ar) (G. Rossi), [erobles@lifia.info.unlp.edu.ar](mailto:erobles@lifia.info.unlp.edu.ar) (E. Robles Luna), [fmontero@dsi.uclm.es](mailto:fmontero@dsi.uclm.es) (F. Montero), [martin.gaedke@informatik.tu-chemnitz.de](mailto:martin.gaedke@informatik.tu-chemnitz.de) (M. Gaedke).

In order to solve this problem we have researched many of the emerging issues in agile development approaches, especially in the requirements engineering field [8]. In this context, user interface prototypes (usually known as *mockups* or *wireframes*), have proven to boost efficiency when capturing requirements of Web Applications [6]. One of their advantages is that they are technically valuable for developers and, at the same time, fully understandable by end-users [9]. In the approach we present detailed mockups (initially originated from User Stories [10]) are used to capture most requirements. We propose using mockups to specify software concepts using high-level models, breaking the linear MDWE workflow into small non-linear cycles in which end-users can actively participate. Then, instead of discarding mockups (as most agile approaches do [11,12]), we transform them into platform-independent UI specifications, and incrementally enrich them to later obtain a set of domain, navigation and presentation models – which can be also iteratively enriched. In this way, mockups serve both as an early requirements gathering tool and as a dedicated starting point for the model derivation process.

The resulting process is primarily based on models, but avoids the *waterfall-like* structure of classic MDWE methodologies. Building the *initial* mockups is the only mandatory step to start modeling, but since anyway mockups are used as requirement artifacts, this does not represent an additional overhead on the process. Once defined, requirements can be quickly modeled over mockups, reducing the requirement-to-software-artifact translation effort. As we show in the paper, this process fits well in agile methodologies like Scrum [13] and, at the same time, makes use of the well-known MDWE infrastructure to avoid *reinventing the wheel*. In face of the broad spectrum of Web Applications types, we decided to focus our approach on data-intensive ones (i.e., *applications whose main purpose is presenting large amount of data to their users* [1], as WebML [14] does), tackling mainly functional requirements. Nevertheless, some non-functional aspects like usability and presentation quality can be evaluated directly using mockups throughout the modeling cycle.

The main contributions of this work are: (1) showing how the traditional MDWE workflow can be improved to allow a less structured specification of concerns, adding agility to the whole process and (2) proposing a metamodel to specify data-intensive Web Applications concepts over user interface prototypes, and derive fully functional MDWE models. We also present a validation to show how the proposed process reduces the development effort in the context of a real-world example.

The paper is structured as follows: in Section 2 we present some background to the approach, in Section 3 we briefly describe an example application that we refer to throughout the paper. In Section 4 we describe our approach in detail and then in Section 5 we detail how it is architected and implemented. In Section 6 we share the results of our field experiences, and in Section 7 we comment related work. Finally, in Section 8 we draw some conclusions and present our future work on this field.

## 2. Background

Digital UI mockup tools are becoming a very popular way of rapidly drafting user interfaces. Their main purpose is to help in discussing UI specifications with end-users, and also to discover and define non-UI requirements in a language that is more familiar to them, as opposed to plain textual specifications [7,9]. In addition, UI mockups work not only as requirements artifacts, but also as general requirements elicitation helpers [15]. Ricca et al. present statistical studies that show how mockups improve requirements gathering in comparison to traditional textual methods, without implying an additional effort in the process [6]. Also, mockups have

been proposed as a successful tool to capture and register *fluid* requirements [16] – those that are usually expressed orally or informally and are an implicit (and usually lost) part of the elicitation process. The importance of capturing additional information or requirements specifications associated to user interface prototypes (which is addressed in this work) is also commented on the work of Ravid and Berry [7].

User Interface prototyping and modeling is an extensively studied field. A plethora of UI modeling methodologies and environments exist [17,18]. UI prototyping tools have quickly emerged in the last years for both desktop (like Balsamiq<sup>1</sup> or Pencil<sup>2</sup>) and Web platforms (like Mockingbird<sup>3</sup> or MockFlow<sup>4</sup>). In this context, the DENIM tool [19] shows an interesting variation of the common *plain* mockup tools, offering interface sketching in different *conceptual* levels and widget drawing by hand. However, these tools are rather focused on building UI sketches bound to be disposed after requirements gathering, or on defining user interfaces in a top-down fashion to generate running applications. In the context of our work, we propose to use mid to high-fidelity UI mockups [20,21], built either with the mentioned tools or by designers in plain HTML. Then we propose to keep them, using their structure as a foundation to specify features like content, navigation or business logic.

Model-Driven Web Engineering (MDWE) approaches like WebML, UWE or OOHDM have a long track proposing improvements in the Web Development field. The main motivation of MDWE is to define the essential aspects of the Web Application using a high level language and then generating the running Web Application automatically, thus promoting more productivity. Similarly, in this approach developers spend their time specifying semantically relevant aspects instead of coding, avoiding coding errors.

We already proposed a technique and tool [22] to include digital User Interface mockups in a Model-Driven process, transforming them into valuable UI specifications that can be used to generate code for several platforms and technologies. By using this background, we also introduced the idea of annotating the obtained models to specify MDWE concepts over the original mockups and then generate final models both for UWE and WebML methodologies [5,23]. In this work, we propose to unify all the methods and tools proposed, describing a detailed UI Mockup-Driven process to develop Web Applications.

## 3. Photo Stock: an example application

In order to explain all the stages in the approach, we will use a Photo Stock website as a sample Web Application. The Photo Stock Web Application was designed analyzing several similar websites, which typically enable users to upload original pictures, sort them in categories/folders and optionally sell their publication permission. Besides these basic functionalities, the application provides the user with a personal blog to post contents related to their photography portfolio or production and it also offers forums where it can participate in discussions.

In the first meeting with our customers, we got some User Stories for a first iteration:

- **User Story 1.** As a User, I want to create, delete and change the name of folders so that I can store and manage my photos.

<sup>1</sup> Balsamiq Mockups – <http://www.balsamiq.com/products/mockups>, accessed: 07-Sep-2012.

<sup>2</sup> Pencil Project – <http://pencil.evolus.vn>, accessed: 07-Sep-2012.

<sup>3</sup> Website Wireframes: Mockingbird – <https://gomockingbird.com>, accessed: 07-Sep-2013.

<sup>4</sup> MockFlow – Online Wireframe Tool – <http://www.mockflow.com>, accessed: 07-Sep-2013.

Download English Version:

<https://daneshyari.com/en/article/550596>

Download Persian Version:

<https://daneshyari.com/article/550596>

[Daneshyari.com](https://daneshyari.com)