

# Taxonomy for Rich-User-Interface Components: Towards a Systematic Development of RIAs

Rosa Romero Gómez, David Díez Cebollero, Susana Montero Moreno,  
Paloma Díaz Pérez, and Ignacio Aedo Cuevas

<sup>1</sup> DEI Lab - Computer Science Department  
Universidad Carlos III de Madrid, Spain

david.diez@uc3m.es, {rmromero, smontero, pdp}@inf.uc3m.es,  
aedo@ia.uc3m.es

**Abstract.** The development of Rich Internet Applications (RIAs) is based on the selection, assembly, and tailoring of Rich-User-Interface (RUI) components. While the user interface design is usually guided by principles, guidelines, and heuristics, there are not mechanisms for systematically selecting RUI components. Moreover, there is a lack of homogeneous classification criteria that hinders the selection of components and increases the relevance of experience designing web applications. To ease the search and the choosing of components by web-developers, this paper presents a taxonomy for classifying RUI components. The development of such a taxonomy has been based on both the study of relevant resources from the UI domain and the opinions of experts.

**Keywords:** Rich Internet Application, component libraries, taxonomy.

## 1 Introduction

Rich Internet Applications (RIAs) are ‘*web applications that offer the responsiveness, rich features, and functionality approaching of that desktop application*’ [2]. These capabilities are achieved by using Rich-User-Interface (RUI) components organized into libraries. These libraries – defined as ‘*reusable components collection for applications development*’ [1]- offer tens of components grouped by diverse criteria and related to different functionalities. Consequently, the search and choosing of the suitable component for a specific problem is not a trivial matter. The component selection process aims at determining which component, among those available, is the most suitable to fulfill a set of requirements of the UI design. The existence of classification schemes is therefore required to be more efficient when using component libraries [1] [3] [6], both during creation and selection of components. However, the experience in the use of RUI component libraries highlights the lack not only of homogeneous criteria for classifying RUI components but also of meaningful terminology. With the purpose of overcoming these limitations, we propose a taxonomy as such a classification artifact that allows categorizing components in a hierarchical way as a manner of easing the selection of RUI components.

This paper presents the taxonomy development process and the taxonomy itself based on both the study of relevant resources from UI domain and the opinions of

Subject-Matter Experts (SMEs hereafter). The rest of the paper is organized as follows. The taxonomy development process is described in section two; this process is divided into two levels – empirical and operational. As the result of such development process, the taxonomy is presented in section three. Finally, conclusions and intentions for further work are drawn in the last section.

## 2 The Taxonomy Development Process

The taxonomy development process refers to its definition by means of the execution of a well-known sequence of activities. In keeping with that, our process, based on the Nickerson's proposal [5], has been divided into two levels – empirical and operational– in order to provide an iterative process that guarantees the appropriate definition of the taxonomy. The empirical phase aims at identifying general characteristics of UI components in order to define a preliminary version of the taxonomy. To achieve this goal, the most direct factors contributing to the quality of the taxonomy [8] - such as corpus, the coverage of the source materials, and its structural model- has been considered. *Standards and guidelines*, *interaction design pattern libraries*, and *development technologies* have been selected because of their relevance and wide use for both UI development and UI design. Due to the disparity of characteristics possessed by complex RUI components – components that may be composed by other RUI components and provide more advanced functionalities, such as direct manipulation-, their categorization was the main challenge during the definition of this initial version of the taxonomy. Finally, this level ends with the definition of a draft taxonomy composed by four main categories – *Controls*, *Widgets*, *Containers*, and *Templates*- and a total of 81 subcategories.

The purpose of the operational phase is twofold. On the one hand, it is conceived to identify missing concepts, misunderstandings, or ambiguous terminology in order to refine our preliminary taxonomy. On the other hand, it allows us to validate the taxonomy and finalize the development process according to the agreement of experts over the relevance of the source materials, the definition of non-ambiguous and meaningful terms, and the consistency of hierarchical relationships throughout the taxonomy. Both activities are based on the feedback provided by SMEs in UI design and web development. Regarding the evaluation technique, we used a mix-questionnaire, i.e. the questionnaire combines both closed-ended questions and open-ended questions. Since *summated rating scales* do not provide concrete values but categories, we used the *median* to identify the agreement of experts: the taxonomy will be considered as valid if the median of experts' opinion is equal or higher than four (agreement level). The operational phase compiles two iterations or evaluations carried out by eleven experts. The initial evaluation suggested that some categories of the taxonomy were ambiguous or misleading. Otherwise, the agreement of experts over the relevance of the corpus of the taxonomy, and the consistency of relationships throughout the taxonomy was validated. Finally, after the second round, the taxonomy has been considered as valid, due to the median of all indicators has achieved the agreement level.

### 3 Taxonomy for Rich-User-Interface Components

The final taxonomy consists of four main categories and a total of 89 subcategories. Its purpose is to classify RUI components in a hierarchical way in order to ease their search and choosing by web-developers. In keeping with this purpose, the definition of the taxonomy was based on intrinsic properties of the components and the common process of choosing components. On the one hand, we identified meta-characteristics that usually serve as basis for the search process: the *structure* – the provision of organization of both interface elements and contents and their relationships to each other- and the *behavior* – the interaction events available- of UI components. On the other hand, the systematic design of the UI [7] is supported by the order of the categories defined in this hierarchical-classification mechanism. The components compiled under *Containers* and *Controls* categories allow respectively establishing the main structure and the basic tasks carried out on the interface. The definition of the input/output information for the users is supported by the components classified under *Widgets* and *Interaction Design Patterns* category:

- **Containers:** it includes the RUI components used to organize the information. These components add support for *modality* – this property enables the developer to scope, or limit, a dialog box’s modality blocking-, *drag and drop* – including moving, copying, or linking selected objects by dragging them from one location and dropping them over another- and default *look and feel characteristics*. Every component must be part of a containment *hierarchy* that has a top-level container as its root [4]. In keeping with that, containers have been divided into two subcategories: *top-level containers* that can hold other UI components; and *intermediate-level containers* that can hold and be held by other UI components.
- **Controls:** it includes the RUI components that allow users to carry out its tasks on the interface. Such category is divided into four subcategories: *imperative controls* used to initiate a function; *selection controls* used to select options or data; *entry controls* used to enter data; and *display controls* used to display the visual representation of information.
- **Widgets:** it includes the RUI components that provide a specific solution to a common design problem. These components use static, *predefined set of look and feel characteristics* in order to ease its configuration. Such category is divided into four subcategories: *entry widgets*; *selection widgets*; *display widgets*; and *navigation widgets*.
- **Interaction Design Patterns:** it includes the RUI components that provide a global solution to a common design problem. These components are usually constructed from aggregates of other RUI components and support setting the *input elements* according to each context of use. The subcategories defined are: *entry patterns*; *selection patterns*; *display patterns*; and *navigation patterns*.

### 4 Conclusions and Future Works

The existence of classification schemes is essential to be efficient when using component libraries. Nevertheless, existing RUI component libraries do not provide

homogeneous classification criteria, resulting in the need of high previous experience as RIA developer to search and choosing suitable components. With the purpose of overcoming these limitations, we have defined a taxonomy that allows us to categorize in a hierarchical way the structural and control elements of the UIs. This taxonomy has been considered as valid after the second round of evaluation according to the judgment of experts.

Further work will be guided to refine our taxonomy and prove its utility. On the one hand, we will carry out the empirical evaluation of both the quality of the taxonomy and its usefulness to select RUI components by means of the provision of a software tool. On the other hand, based on specific web-based interactive design scenarios, users testing will be carried out in order to assess the usefulness of the taxonomy to select RUI components.

## Acknowledgements

This work has been partly supported by both the RIA – User Interface with Patterns and Components Reuse project funded by the Regional Government of Madrid (CAM) and the Universidad Carlos III de Madrid (UC3M) and the urTHEY project (TIN2009-09687) funded by the Ministry of Science and Innovation (MICINN) of the Government of Spain.

## References

1. Curtis, N.: *Modular Web Design: Creating Reusable Components for User Experience Design and Documentation*. New Riders, Indianapolis (2009)
2. Deitel, P.J., Deitel, H.M.: *Ajax, Rich Internet Applications and Web Development for Programmers*. Deitel Developers Series (2008)
3. Frakes, W.B., Pole, T.P.: An empirical study of representation methods for reusable software components. *IEEE Transactions on Software Engineering* 20(8) (1994)
4. *Java Look and Feel Design Guidelines*. Sun Microsystems Inc. Version 2.0 (2001), <http://java.sun.com/products/jlf/ed2/book/index.htm> (retrieved November 15, 2011)
5. Nickerson, R.C., Varshney, U., Muntermann, J., Isaac, H.: Taxonomy Development in Information Systems: Developing a Taxonomy of Mobile Applications. In: *17th European Conference on Information Systems* (2009)
6. Pressman, R.S.: *Software Engineering: A Practitioner's Approach*. McGraw-Hill, New York (2005)
7. Stone, D., Jarrett, C., Woodroffe, M., Minocha, S.: *User Interface Design and Evaluation (Interactive Technologies)*. Morgan Kauffman Series (2005)
8. Vogel, C., Powers, J.: *Quality Metrics: How to Ensure Quality Taxonomies*. Information Today (2000)