End-User Support for Information Architecture Analysis in Interactive Web Applications

Luis A. Rojas and José A. Macías

Escuela Politécnica Superior, Universidad Autónoma de Madrid,
Tomás y Valiente 11, 28049 Madrid, España
luisalberto.rojas@estudiante.uam.es, j.macias@uam.es

Abstract. Information Architects analyze conceptual models and descriptions concerning non-functional requirements that will be later on used by Analysts and Software Engineers to design software artifacts. However, this flow of information is sometimes difficult to automate due to conceptual knowledge differences in the information processed by each expert. To facilitate this flow, in this paper we propose a CASE tool called InterArch. Our tool bridges the gap between both professionals, and it automatically generates UML diagrams for Analysts from the conceptual diagrams created by the Information Architect.

Keywords: Information Architecture, End-User Development, Analysis and Design of Interactive Web Systems.

1 Introduction

The Information Architecture (IA) is defined as the science of structuring, organizing and managing information, where labeling, findability and usability are specifically important for a good information performance [2]. Its main objective is to facilitate, as much as possible, the comprehension and assimilation of the information [1], as well as to execute tasks in user-defined information spaces [6].

The Information Architect attempts to find a bridge between the conceptual knowledge proposed by users in the problem domain and the design information that Software Engineers require to design the final interface in the solution domain. This gap frequently means that some degree of interoperability and alignment between the output generated by the Information Architect and the input expected by the Software Engineer is necessary in order to accomplish the whole software design. Furthermore, automating this flow of information could benefit from reducing both time and costs in software projects, also allowing each expert to focus on their own work and optimizing the transfer of knowledge among both professionals [5].

In order to address this problem, we have designed a CASE tool called InterArch, which allows experts in the problem domain to focus on content development and browsing. This way, our tool automatically generates UML content classes for Software Engineers from the conceptual models crated by Information Architects, thus automatically specifying the right elements for the solution domain.

2   Proposed Solution

Most common products that Information Architects have to create to carry out the IA of an interactive web applications correspond to *blueprints*, *wireframes*, *content models* and *controlled vocabularies* [2, 6]. From all this products, *content models* are particularly important for Analysts and Software Engineers, and they are very likely to be automatically processed in order to generate content classes and objects that define the solution in the application domain. This is the reason why we have focused on these essential products in favor of an explicit automation to provide useful input for the Software Engineer.

To address this challenge, our tool is based on two essential points: first, InterArch enables Information Architects to concentrate on conceptual-design tasks in the problem domain. Second, based on the initial analysis carried out by Information Architects, InterArch automatically generates UML diagrams for Analysts and Software Engineers, specifying elements that have a direct correspondence with the information content. UML information is generated in a transportable text-like format called XMI[1] that improve interoperability and can be processed by a great deal of existing CASE tools in order to accomplish the rest of the phases in the software project lifecycle.

2.1   A CASE Tool for the Information Architect

InterArch comprises a set of processes that are responsible for the management and transformation of information models in a visual environment for the IA analysis. These processes include: visual modeling of the conceptual elements required by the information professionals, transformation of the visual model into an intermediate model, and textual-transportable generation of the analysis of the information using UML diagrams. In a nutshell, these processes aim to take the input – i.e., the visual diagrams created by the Information Architect, and generate the output comprising UML diagrams for Analysts and Software Engineers.

The main idea behind these architectural components is to enable Information Architect to work in the visual processing of diagrams in a transparent way, but also featuring an interpretation overlay that recognizes the different correlations between such analysis diagrams and the UML classes required by Software Engineers. The transformation of the visual model is achieved through a set of relations and association rules which are applied to the conceptual model in order to generate the corresponding UML diagrams in XMI format.

Figure 1 depicts the user interface of InterArch, whose main parts are labeled with A, B and C. Part A includes a tool bar that implements functionalities concerning files, edition, format and style and others. Part B represents the main authoring environment used by the Information Architect. Finally, part C shows the icon library to enrich the visual composition of diagrams developed by the Information Architect.

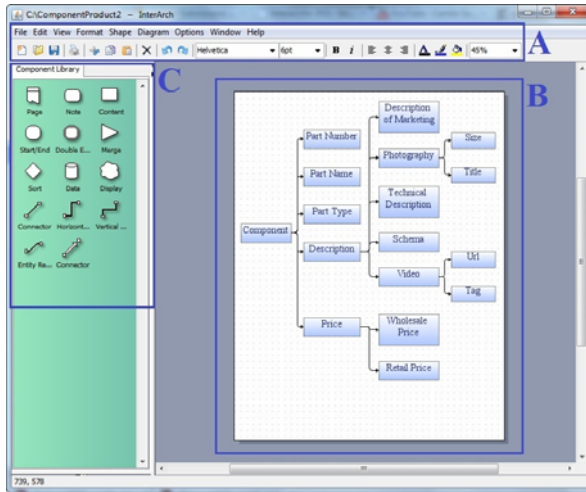[1] XMI is the OMG-defined standard for the exchange of UML diagrams, www.omg.org

Fig. 1. InterArch User Interface

2.2   A Worked Example

In order to describe our tool in detail, we provide a specific example using InterArch. Let us suppose that the Information Architect is working on a content model including information about the composition and prices of products in order to carry out the IA analysis of an on-line shop. This situation corresponds to the diagram depicted in Figure 1.
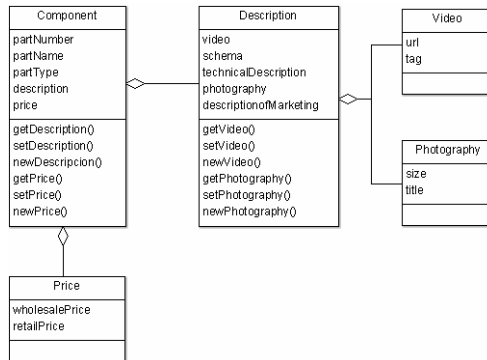

Fig. 2. UML Class Diagram generated by InterArch

To generate UML diagrams based on IA diagrams, transformation rules are applied in order to find out meaningful relationships. Carrying on with the example (see Figure 1), element **Component** will be first generated as a main class, also considering the descendant elements that will be processed by applying the following rule: terminal elements, and direct descendants from the main element, will become

attributes for class **Component**. If these elements also have descendants, they will directly become new classes related to **Component**. This rule is recursively applied for all descendant elements in order to generate new classes, attributes and relationships. This way the application of this rule generates, for instance, a class **Component** with five attributes: **partNumber**, **partName**, **partType**, **description** and **price**. As for the class methods, three methods for each attribute that represents an aggregate class are generated by default (get, set and new). Figure 2 shows the final visualization of the UML classes generated including all the elements.

3   Conclusion and Future Work

In this paper, we propose an approach that provides an automatic solution to bridge the gap between high-level conceptual representations of content and the UML analysis and design classes needed to implement such content.

Our research is based on the EUD (End-User Development) paradigm [3, 5], where the main idea is to help end-users create or customize applications easily. More specifically, our work is based on EUSE (End-User Software Engineering), whose aim is to involve non-expert users, as often happens with Information Architects, in particular issues of the development process in software analysis and design [4].

As future work, we expect to provide new features, such as the generation and customization of rich design elements like design and content components. Also, we expect to carry out a usability evaluation, in order to study the trade-off between ease of use and expressiveness by involving professional Information Architects.

References

1. Elaine, G.T.: Information interaction: Providing a framework for information architecture. Journal of the American Society for Information Science and Technology 53(10), 855–862 (2002)
2. Erlin,Yunus, Y. A., Rahman, A.A.: The evolution of Information Architecture. In: International Symposium on Information Technology, ITSim 2008. , vol.4, pp.1–6 (2008)
3. Klann, M., Fit, F.: End-User Development. D1.1 Roadmap. In: Proceedings of the End User Development Workshop at CHI 2003 Conference, Ft. Lauderdale, Florida, USA, April 5-10 (2003)
4. Lieberman, H., Paternò, F., Wulf, V.: End-User Development. Human Computer Interaction Series. Springer, Heidelberg (2006)
5. Macías, J.A., Castells, P.: Providing end-user facilities to simplify ontology-driven web application authoring. In: Interacting with Computers, vol. 19(4), pp. 563–585. Elsevier, Amsterdam (2007)
6. Morville, P., Rosenfeld, L.: Information Architecture for the World Wide Web, 3rd edn. O'Reilly Media", O'Reilly Media Inc (2006)