Science of Computer Programming 78 (2013) 2282–2291

Contents lists available at ScienceDirect

Science of Computer Programming

journal homepage: www.elsevier.com/locate/scico

Bridging the gap between information architecture analysis and software engineering in interactive web application development

Luis A. Rojas, José A. Macías *

Escuela Politécnica Superior, Universidad Autónoma de Madrid, Tomás y Valiente 11, 28049 Madrid, España, Spain

ARTICLE INFO

Article history:
Received 15 January 2012
Received in revised form 11 July 2012
Accepted 26 July 2012
Available online 20 August 2012

Keywords:
Human–computer interaction
Information architecture
End-user development
Software engineering

ABSTRACT

Web development teams comprise non-computer experts working on the conceptual modeling of non-functional aspects in software applications. Later on, such conceptual information is processed by analysts and software engineers to face the technical phases of the software project. However, this information transfer is often difficult to automate since the information processed by the different professionals involves different abstraction levels, as well as important cost and effort that need to be considered. The main aim of this research is to minimize these problems by increasing automation and interoperability in the development of interactive web applications. To take up this challenge, we have created and evaluated a tool that aims at bridging the gap between the conceptual definitions of web contents – i.e., the information architecture, and the UML elements for analysis and design required by software engineers, connecting functional and non-functional information to achieve the rest of technical activities during the software development process.

1. Introduction

Information architecture (IA) is a recent paradigm that has been gradually introduced in most web development projects today. IA is defined as the science of structuring, organizing and managing information, where the usability plays an important role in the solutions created [1]. IA is directly involved in the development of website, mobile devices, vending machine and electronic games interfaces, to cite a few. Its main objective is to facilitate the processing and assimilation of information, as well as the understanding of tasks performed by users in a defined information space [2]. The way people interact with digital information environments is directly influenced by the information architecture [3].

The information architect is the person in charge of the IA. S/he works in the early phases of interactive web application development, attempting to connect the conceptual knowledge supplied by users and the technical information (i.e., analysis and design) needed by software engineers responsible for implementing the final web application. However, it is very common that the roles of information architect and software engineer rarely match, as the information architect may have a non-technical profile more oriented to conceptual tasks or ergonomics. This makes it necessary to ensure some degree of interoperability and alignment between the output generated by the information architect and the input that the software engineer needs. If this flow of information can be done in an automatic way, time and effort can be drastically reduced in the software project, allowing each expert to concentrate on the task according to her/his knowledge, and so minimizing the time of the information transfer between both kinds of professionals [4].

The aim of this paper is to overcome this problem and bridge the gap between the tasks performed by the information architect and the ones achieved by the software engineer. To carry out this task, we propose a CASE (Computer Aided Software Engineering) tool called InterArch [5] (Interoperable Information Architecture), which allows experts in the

* Corresponding author. Tel.: +34 914976241; fax: +34 91 4972235.
E-mail addresses: luisalberto.rojas@estudiante.uam.es (L.A. Rojas), j.macias@uam.es (J.A. Macías).

0167-6423/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.scico.2012.07.020

problem domain to focus on content analysis and navigation while the tool automatically generates UML classes for software engineers, implicitly supplying elements in the solution domain.

Specifically, our research is based on the following objectives:

- Generate analysis and design information for analysts and software engineers from the conceptual representations of contents earlier created by the information architect.
- Build an easy-to-use CASE tool for the information architect that enables to automatically generate analysis and design information for analysts and software engineers from the conceptual descriptions created by the information architect, and so bridging the gap between the initial project activities, in the problem domain, and the technical development activities, in the solution domain.
- Evaluate the usability of this tool by means of a user experiment, in order to obtain initial feedback for improvement through an iterative and incremental end-user-centered development process.

This paper is structured as follows. Section 2 introduces the related work. Section 3 presents our approach in detail. Section 4 describes the transformation rules used in our approach. Section 5 provides a use case to show the functioning of our tool in detail. Section 6 reports on an evaluation with real users to measure the usability of our approach. Finally, Section 7 discusses conclusions and future work.

2. Related work

There are a great variety of tools for the creation of diagrams representing the information architecture [6], and also for analyzing and evaluating the information architecture in websites [7]. These tools correspond to desktop and online software applications commonly used by information professionals to draw *blueprints* and create *wireframes* and *content models*, such as Axure, Visio Professional, OmniGraffle, Denim, ConceptDrawPro, SmartDraw, Pencil Project, MockFlow, iPlotz, Pidoco, Lovely Chart, Mockingbird and Lumzy, to cite a few. These tools include libraries comprising graphical elements for web prototyping, which enables managing and publishing information elements as well as incorporating new graphical components. Commonly, some of these tools include annotations, footnote facilities, collaborative authoring and dynamic prototyping.

Other approaches address non-functional requirements representation like informal architecture documentation, UML diagrams, and Architecture Description Languages (ADLs). These tools provide abstraction from implementation details, as well as data structures and relationships between different components [8]. Although these approaches present some drawbacks in connecting architectural descriptions and implementations, this has been solved in [9] by presenting an ADL-based solution that supports the modeling of system architectures at different levels of abstraction, linking architectural concepts to different technologies, immediate conflict detection, and continuous synchronization of both architecture and implementation. However, none of the approaches included in [9] concrete explicit relationships between IA content models and UML class diagrams, or even mechanisms to represent non-functional information and transform it into UML descriptions to be exploited in the software project.

On the other hand, online tools are becoming more popular than desktop stand-alone versions, due to their availability and free-of-charge facility. However, online tools are generally less expressive and complete than desktop versions in terms of functionality. On the other hand, most common tools experience difficulties in connecting the output generated by the information architect and the input needed by the software engineer. This problem has been traditionally addressed by generating different graphical formats and HTML code to interchange information. However, this solution does not consider issues related to semantic analysis included in IA diagrams, and it lacks interoperability among different professionals working together, making it difficult to manage and share the knowledge generated by different tools and professionals in the software project.

3. Proposed solution

Generally, it is difficult to identify the operational limits of the information architecture, sometimes requiring the use of different tools and standards. However, it is possible to summarize the most common products that the information architect creates to carry out the analysis of the information architecture in interactive web application. Those are *blueprints*, *wireframes*, *content models* and *controlled vocabularies* [1,2]. These products provide important knowledge regarding the analysis, organization, managing and structuring of information for the professionals involved in the development of web projects. However, for all these products, content models are particularly transcendental for analysts and software engineers, as they represent non-functional aspect of the web application and are susceptible of being automatically processed to generate content classes and object diagrams that will define the software application in the solution domain. In fact, our approach is focused on these core elements for automating the output of the IA analysis.

This way, we have designed a CASE tool called InterArch that is based on two essential principles. First, since the information architect usually has a non-technical profile, more oriented to information design and organization, InterArch allows the information architect to concentrate on conceptual analysis tasks in the problem domain. Second, based on the initial analysis carried out by the information architect, InterArch automatically generates UML diagrams for analysts and software engineers, identifying elements that have a direct correspondence with class diagrams and content objects used by software professionals. In order for the information to be processed by any common CASE tool and provide continuity for