

Chapter 3

Integrating Tagging Software in Web Application

Karan Gupta

University of Delhi, India

Anita Goel

University of Delhi, India

ABSTRACT

Tag software is included in web applications to facilitate categorization and classification of information. Generally, freely available tag software is adapted or new code written to incorporate tagging. However, there is an absence of requirement and design document for tagging, even academically. It becomes difficult to know the features that can be included in tag software; also, not all features may be required. This chapter presents a framework for integration of tag software in web applications. The framework has four components corresponding to phases of the software development lifecycle. For requirement, a weighted requirement checklist is presented to ease requirement selection. A metric, software estimation, is defined for quantifying selected requirement. A logical design defined for design phase displays interaction of entities with users. For development, best mechanisms are suggested to web applications. Software engineering artefacts are provided to help during testing. A case study is presented where estimation and design is applied to freely available tag software.

INTRODUCTION

The popularity of Web applications has increased tremendously over the last decade. Nowadays, Web applications are available for almost every field, like, medicine, sports, news, and education. With the increasing popularity of Web application, large amount of information is stored in Web in various forms like text, audio and video. To ease burden of users for managing large amount of information, Web applications incorporate tag software.

Tag software incorporated in a Web application improves the search process for a resource. Tag software allows the user to add keywords (also known as tags) to a resource. The resource for tagging may

DOI: 10.4018/978-1-5225-5384-7.ch003

be a video, audio, blog, books etc. Tags added to a resource, generally, describe the resource but can also define its type, its use, pros and cons or something entirely different. Tag software is used in a variety of Web applications, like, products available for sale in an online retail site and albums in a music site.

Several options exist to include tagging functionality in a Web application, like, using free tag software, adapting a freely available tagging code or writing a new code for tagging. The most commonly followed approach is to adapt free tag software to suit the needs of a Web application. The code is modified and customized to match the appearance of Web application. Alternatively, Web applications may write their own new code. Although software exists in form of free tag systems and freely available code, there is no mention of a document stating requirements and design of the tagging functionality. Generally, tagging functionality is integrated on-the-fly depending on the whims and fancy of developer and stated requirements of a Web application.

However, the use of ad-hoc approach increases the burden of Web application owner and developer during the development of tag software. Since, a requirement specification document for tag software is not available, integrating tag software in a Web application becomes difficult. When creating new tag software for integration, the task of requirement elicitation has to be repeated every time, a new. Moreover, the requirement elicited during the requirement specification phase, may not be complete as the Web application has limited or little knowledge about the features of tag software. If existing tag software is to be integrated in a Web application, it becomes a cumbersome task to identify if complete features are provided by the existing tag software or some specific features have been skipped. Also, some features that may be useless for the Web application may remain undetected, which shall affect the size and performance of the tag software.

Due to the absence of availability of design document for tag software, the task of updating the tag software becomes difficult. When using existing tag software, the developer needs to arrive at the architecture of tag software using reverse engineering, i.e. deriving design from the code. This task is required to be performed when in existing tag software, a new feature is required to be added, or an existing one is to be modified or deleted. Also, there is a need to understand the design of tag software in case of error diagnosis. When writing new tag software, the developer needs to start from scratch and repeat the design process every time new code is written.

Here, the authors provide a framework for easing integration of the tag software into a Web application in accordance with various phases of the software development lifecycle. The framework is an enhanced version of our earlier framework, presented in (Gupta & Goel, 2013). The enhanced framework helps the Web application owner and the developer in understanding the tag software. In the enhanced framework, there are four components: (1) Tagging_Requirement, (2) Tagging_Design, (3) Tagging_Development, and (4) Tagging_Test. Each component has a specific task like Tagging_Requirement component performs the task of requirement generation. The four components interact with the Web application owner and the developer to generate and integrate the tagging software into Web application. Each component is divided into sub-components so as to ease the completion of its task.

The enhanced framework is used by both the developer and the owner of Web application during integration of tagging software into a Web application. The developer uses the enhanced framework to understand the structure of the tagging software. The owner of Web application gets to know the different kinds of users accessing the tagging software as well as the different kinds of features provided by the tagging software. The owner of the Web application is able to select these requirements.

The enhanced framework uses various software engineering artefacts developed in (Gupta & Goel, 2014) to ease the task of integration of tag software into Web application. During the requirement elici-

20 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/integrating-tagging-software-in-web-application/203416

Related Content

Enhanced Utility in Search Log Publishing of Infrequent Items

S. Belinsha and A.P.V. Raghavendra (2014). *International Journal of Information Technology and Web Engineering* (pp. 65-77).

www.irma-international.org/article/enhanced-utility-in-search-log-publishing-of-infrequent-items/115935

Multi-Objective Optimization-Oriented Resource Allocation in the Fog Environment: A New Hybrid Approach

Sonti Harika and B. Chaitanya Krishna (2022). *International Journal of Information Technology and Web Engineering* (pp. 1-25).

www.irma-international.org/article/multi-objective-optimization-oriented-resource-allocation-in-the-fog-environment/297969

The Role of Electronic Signature in the Third Millennium of the Development of International Trade Transactions

Amir Bagherian, Yosef Ebrahimi Nasaband, Hassan Heidari and Mahmoud Ebrahimi (2016). *Web Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 170-183).

www.irma-international.org/chapter/the-role-of-electronic-signature-in-the-third-millennium-of-the-development-of-international-trade-transactions/137345

Schema Change Operations for Full Support of Schema Versioning in the XSchema Framework

Zouhaier Brahmia, Fabio Grandi, Barbara Oliboni and Rafik Bouaziz (2014). *International Journal of Information Technology and Web Engineering* (pp. 20-46).

www.irma-international.org/article/schema-change-operations-for-full-support-of-schema-versioning-in-the-xschema-framework/115933

Algorithm for Secure Hybrid Cloud Design Against DDoS Attacks

Akashdeep Bhardwaj and Sam Goundar (2018). *International Journal of Information Technology and Web Engineering* (pp. 61-77).

www.irma-international.org/article/algorithm-for-secure-hybrid-cloud-design-against-ddos-attacks/209721