Requirement Estimation and Design of Tag software in Web Application

Karan Gupta, Department of Computer Science, University of Delhi, New Delhi, Delhi, India

Anita Goel, Department of Computer Science, Dyal Singh College, University of Delhi, New Delhi, Delhi, India

ABSTRACT

Tag software is included in a web application to facilitate categorization and classification of information. Generally, freely available tag software is adapted, or new code is written to incorporate tagging functionality. Since there is an absence of requirement specification and design document for tag software, even academically, it becomes difficult for the user to know about the possible features that can be included in the tag software. The user has to search for those features to be able to implement them in the software. So, there is a need that the user is made aware of the features available. Moreover, not all the features are relevant for the user; hence, there is a need for some kind of mechanism to ease the decision process. This paper presents - (1) a design for tag software, and (2) categorization of requirements of tag software in a web application. The design helps the developer during updating and analysis. The logical view of design displays interaction of entities and sub-entities with users. A weighted requirement checklist is presented which segregates features in three categories based on their popularity. This eases the task of selecting the requirement of tag software for the user. A metric, software estimation, is defined for quantifying selected requirements. A case study of freely available tag software is presented, in which estimation and design is applied.

Keywords: Integration, Logical View, Tag Software, Web Application, Weighted Table

INTRODUCTION

Web applications have become enormously popular during the last decade. The domain of web applications has spread to various fields, like, medicine, sports, news, and education. With the increasing popularity of web application, large amount of information is stored in web applications in various forms like text, audio, video etc. To ease burden of users for managing large amount of information, web applications incorporate tag software. Tag software is being used by web applications to manage large amount of information. Tagging improves search within a resource. Tag software allows the user to add keywords (also known as tags) to a resource. The resource for tagging may 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 application, 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, there is non-availability of a requirement specification document for tag software; 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 in a web application.

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.

In this article, we focus on creation of a design document and a specification for prioritizing requirements of tag software in web application. The document and the specification are to be used during the development of the tag software. Both the design and the specification help the web application owner and the developer in understanding the tag software.

Here, we provide a design for tag software in a web application in form of logical view which acts as a design document. The logical view facilitates the web application owner and the developers in understanding the architecture of tag software. The logical view is depicted for two main building blocks of tag software – tag and resource. For the creation of logical view, a group of entities, for both tag and resource, have been identified. The entities are further divided into sub-entities which act as a container for operations that are performed in the tag software. Also, the interaction of entities with actors of tag software has been defined in the logical view.

Also, a set of weighted requirement checklists is provided for tag software in web application. The weighted requirement checklist aids developer and web application owner in deciding requirement for tag software in web application. The weighted requirement checklist depicts different components of tag software – tagging home, tagging parameters and tagging dashboard. The requirement checklists, presented by Gupta and Goel (2012), are used as basis for developing weighted requirement checklists. We conducted a survey on two freely available tag software and six web applications.
Related Content

Analysis of Quality of Service Routing Algorithms
[www.igi-global.com/article/analysis-quality-service-routing-algorithms/4036?camid=4v1a](www.igi-global.com/article/analysis-quality-service-routing-algorithms/4036?camid=4v1a)

SWAMI: A Multiagent, Active Representation of a User’s Browsing Interests
[www.igi-global.com/article/swami-multiagent-active-representation-user/37586?camid=4v1a](www.igi-global.com/article/swami-multiagent-active-representation-user/37586?camid=4v1a)

Virtual Machine Placement Strategy for Cloud Data Center
[www.igi-global.com/chapter/virtual-machine-placement-strategy-for-cloud-data-center/140829?camid=4v1a](www.igi-global.com/chapter/virtual-machine-placement-strategy-for-cloud-data-center/140829?camid=4v1a)
Demand-Driven Algorithm for Sharing and Distribution of Photovoltaic Power in a Small Local Area Grid