Collaborative Social Tagging for eBook using External DSL Approach

유환수, 김성환
서울시립대학교 컴퓨터과학부
e-mail : rayryu7@uos.ac.kr, swkim7@uos.ac.kr

Hwan-Soo Yoo*, Seong-Whan Kim**
*Dept. of Computer Science, University of Seoul

Abstract

We propose a collaborative social tagging for eBook using external DSL approach. The goal of this paper is (1) to provide DSL by which authors can write HTML5 rich contents ebook and tag resources, (2) to make users enhance book by tagging resources easily, (3) to make readers read rich book easily regardless of their devices types, (4) to provide ebook resources of RESTful address style by which other system can identify self-descriptive resources of book. To achieve the goal, we provide Bukle DSL language by which author and users can author and enhance ebook with ease. As a domain-specific language Bukle provides a simple yet expressive language for authoring and tagging books that would otherwise be more difficult to express with a general purpose language. Further work includes visual DSL approach and tools by using that the unskilled users could tag book easily. In order that future work also includes text-to-visual DSL transform engine. UX research is also required to tag and to author book. To tackle the above questions we are looking at using visual notation focusing visual syntax.

1. Introduction

Project Gutenberg [1] is a collaborative project to distribute information as the e-book by collecting assets of full text of public domain books. It is currently claimed over 45,000 items in its collection and newly registered more than 50 books in every weeks. We borrow idea from folksonomy and social tagging to add rich media into book domain.

A folksonomy [2] is known as ‘classification by the people (Folk+order+nomos)’ which is a new classification system that divides according to ‘tag’ instead of ‘directory’ which is traditional classification system. The existing classification system of taxonomy is different from folksonomy. Folksonomy systemizes unit of information by individual members giving a meaning to information. This paper uses social tagging by which users can add resources to existing books of Project Gutenberg and do that when authors write a new book. By using social tagging users can participate in the project voluntarily and the book may gradually evolves to mature rich contents e-book by social user.

DSLs are referred to as either domain specific language or just simply domain language but are often referred to as domain specific language by academic community. An advantage of using DSL is when domain’s complexity if high, development that have used DSL have high efficiency compared to development investment[3, pp. 20–22] . DSL, on the other hand has intrinsic domain knowledge and therefore is easier to store, verify, and reuse than applications make with general purpose language. This paper’s domain specific language Bukle processing workflow are divided into three phases for effectiveness and necessity of real-time HTML translation. In inception phase, legacy ebooks such as ebooks of Gutenberg project are translated to Bukle-formed book. In elaboration phase, HTML 5 model and social tag are used to translate Bukle to HTML5 ebook. In this phase, real-time translation is needed so that Bukle is merged with social taggers’ social tag. Then, merged ebook will be translated to HTML5 ebook. In improvement phase, readers will tag the ebook to enhance it when reading books.

This paper is organized as follows: in section 2, we describe motivations such as the volunteer’s participation of Gutenberg project, changing the collective intelligence using folksonomy, prevalence of HTML5 and uniform interface for resources of REST. Section 3 provides goals and objectives of this paper. The Bukle language, language workflow and example are describe in section 4. Section 5 includes the related work in the field. Finally, section 6 presents conclusions and future work.

2. Related work

2.1 *Ebook*

The ebook includes text-based digital contents as well as multimedia. Contemporary ebook is not a plain text book but a playing book or multimedia book combined elements

of multimedia. The existing paper book gradually being looked away because of the inconvenience of portability but ebook has grown over rapidly through the prosperity of various devices such as smartphones and smartpads. The type of e-book format includes ePub[4], PDF, XML, HTML, Flash and ebook application. ePub (electronic publication) was adopted in September 2007 for standards established by the International Digital Publishing Forum. ePub consists of the structure such as OCF (Operating Cash Flow), OPF (Open Packaging Format), OPS (Open Publication Structure) and OCF rules for the logical configuration information. Even if ePub 3.0 supports HTML5 and CSS3, it does not supports browser rendering of its contents. ePub model must be transformed to HTML5/CSS3/JavaScript model to render its contents in browser properly. DSL technology is used when target ebook reading platforms are adapting browser rendering. This paper's Bukle language is used in smartphone's app [5][6] to make author to write and decorate book easily. It was also introduced in NY Times [7] this year. To transform ePub to HTML5, Vincent le Chevalier et al[8] used face recognition algorithm and three process of correlation sub-system to guarantee that a transformed HTML5 ebook has fidelity quality.

2.2 *Folksonomy*

Daylamani Zad [9] experimented and gave results that collected data from users using both folksonomy and MPEG-7 tools to annotate and retrieve media. They suggested a conceptual model for each type of tool that illustrates the tag usage. They argued that this conceptual models are then exploited in the development of an MPEG-7 profile for multimedia annotation communities. Kiu et al [10] presents taxonomy and folksonomy integration algorithm, namely TaxoFolk to integrate the folksonomy into a taxonomy to enhance knowledge classification and navigation. Peleteiro et al [11] present an algorithm for multimedia content recommendation using folksonomy. To measure its performance, they have developed a prototype, TAGGEMENDOR.

2.3 *DSL-based Development*

DSLs are languages that define the jargon of a particular class of problem domains or set of domain aspects. Executable DSLs hide software implementation [12].

DSL is referred to as either domain specific language or just simply domain language but is often referred to as domain specific language by academic community. It is a limited language that is concentrated to specific problem domains by appropriate abstraction and notations. It is a problem-centric language [13] built for one particular task in mind. Development teams use same vocabulary to represent program module. For example, if there is a requirement for bond trading module within problem domain, same vocabulary is used during code writing [3, p. 8].

Kramer [14] researched mobile development DSL by supporting functions such as variables, functions, widgets, and events for deployment to iPhone, Android, and Windows Mobile phones. Funk [15] have researched using Xtext [16] and Xtend [17] for DSL to provide easy tool since image, sound, voice, and video was hard to handle for designers or content producers using existing complex tools.

Furtado [18] have defined method on how to make games that operate under client using DSL, SPL [19], and FODA [20]. Maximilien [21] suggests that current web mashups provide a flourishing and innovative Web, but it also states that it raises new challenges, including that current mashups are point solutions, lacking principled approaches to architecture and design; and many similar mashups use the same APIs but share nothing else in common. Therefore, by using DSL, developers can also use mashups to solve Web integration and service composition problems. Masuch [22] integrated DSL into scenario-based game.

3. Motivation

Project Gutenberg is a project to distribute information as the e-book by collecting assets of full text of public domain books. This project is founded in 1971 by Michael S. Hart is named after Johannes Gutenberg who rapidly expanded the knowledge by inventing the printing. The aim of this project is to create a virtual library that users can download and read ebook saved in the Internet as e-text and many volunteers typed published book with the project using the Internet. It is currently claimed over 45,000 items in its collection and newly registered more than 50 books in every weeks. The e-book registered on project mostly consists of the west literature and in addition to literary works such as novels, poems, short stories, and drama, the dictionaries, periodicals, and cooking book are included. In addition, some of the audio files and sheets of music files are stored. We borrow idea from folksonomy and social tagging to add rich media into book.

A folksonomy [2] is known as 'classification by the people (Folk+order+nomos)' which is a new classification system that divides according to 'tag' instead of 'directory' which is traditional classification system. The existing classification system of taxonomy is different from folksonomy. Folksonomy systemizes unit of information by individual members giving a meaning to information. The model of this paper uses social tagging by which users can add resources to existing books of Project Gutenberg and do that when authors write a new book. By using social tagging users can participate in the project voluntarily and the book may gradually evolves to mature and rich contents e-book by social user.

REST (Representational State Transfer) [23, pp. 4–10] is software architecture for a distributed hypermedia system like World Wide Web. Uniquely identified URL is assigned to resources by common grammar. The architectural properties of REST are (1) the separation of clients from server, (2) statelessness by which no client context being stored on the server between request, (3) using a uniform interface to identification of resources and self-descriptive messages.

4. Goals and Objectives

The goal of this paper is (1) to define the model of evolvable social resources taggable book, (2) to provide DSL by which authors can write HTML5 rich contents ebook and

tag resources, (3) to make users enhance book by tagging resources easily, (4) to make readers read rich book easily regardless of their devices, (5) to provide ebook resources of RESTful address style by which other system can identify self-descriptive resources of book.

5. Collaborative Social Tagging for eBook with Evolution Support using External DSL Approach

5.1 *Overview*

The author of book or volunteer of project Gutenberg who know the programming language, HTML5, CSS3, or JavaScript are unusual. We provide easy DSL (Domain Specific language) [12] so called Bukle by which author and users can author and enhance ebook with ease.

5.2 *Bukle phase*

Bukle language processing workflow are divided into three phases. In inception phase, legacy ebook such ebooks of Gutenberg project are translated to Bukle-formed book. Template such as transform template, theme and effect is also used to translation.

In elaboration phase, HTML 5 model and social tag are used to translate Bukle to HTML5 ebook. HTML5 model template are consists of HTML5 boilerplate, CSS3 boilerplate, JavaScript boilerplate and theme. In this phase, real-time translation is needed so that Bukle is merged with social taggers' social tag. Then, merged ebook will be translated to HTML5 ebook.

In improvement phase, readers will tag the ebook to enhance it when reading books. In that case, reader will be not only passive user but also active contents producer.

5.3 *Bukle language design*

To meet the language requirements, we design Bukle as Fig. 1 오류! 참조 원본을 찾을 수 없습니다.. Binder is ebook comprised of Bukle language. It is later converted to browser renderable ebook that contains HTML, CSS3, JavaScript, link and media resources.

bukleΔ	→	**binder**
binder	→	**chapter***
chapter	→	**stmt+**
stmt	→	**strucKle+ | headingKle+**
		| subHeadingKle+ | linkKle+
		| text+
strucKle	→	**struckleDelimiter | structType**
headingKle	→	**heading**
internalLinkle	→	**linkle**
struckleDelimiter	→	**"@@" structType**
structType	→	**chapterKle | sectionKle**
chapterKle	→	**"chapter:" constString**
themeKle	→	**"theme:" constString**
linKle	→	**"[@" linkType ":" (title | url**
		| fileName) caption
meta	→	**'@" key ":" value**
caption	→	**"{" text "}"**
……		

Fig. 1 Partial BNF for Bukle

Partial language syntax and usage is as Table 1 오류! 참조 원본을 찾을 수 없습니다.

Table 1 Language syntax

Category	Description	Example
@@	• Structural markup Separator markup for item when using multiple chapter into one file. Designate theme of books.	@@chapter:introduction-1 @@theme:Greenday
@[meta key:meta value]	• Meta data Store key, value pair of item. It is used by preprocessor and template when publishing	@section: Introduction @section: Participants @parts: English army
[meta key]	• substitution within [] pair, key is substituted to value of key	[#section]
#	• Heading	# Significance of the American Revolution
##	• Sub-heading	## The world
link	• [@linkCategory:title or file name]{Link Caption}	[@term:George_III_of_the_United_Kingdom]{King George III}
External link	• [@linkType:linkURL]{Link Caption}	[@video: http://youtu.be/cTZAzVzFShw]{Pride and Prejudice Movie}
Image link	• [@image:FILENAME]{Image Caption}	[@image:BostonTeaParty.jpg]{Boston Tea Party}
Sound link	• [@music:FILENAME]{Music Caption}	[@sound:PianoForte.mp3]{Piano Forte}
Video link	• [@video:FILENAME]{video Caption}	[@video:CrossingOfTheDelawareRiver.mpg]{crossing of the Delaware River }
Background	• [@bg:FILENAME]{description}	[@bg:Heyday.jpg]{heyday}

@@ is a structural markup which is used when users pack multiple item into one file.

is used for text heading. Which is similar to '===' of Markdown [24]. ## is used for sub-heading. Blank line is used for separating the paragraphs.

@ is used for meta key/value pair.

The structure of link and external link is [@linkCategory]{Link Caption}. The link category contains image, music and video.

We also design RESTful address style of ebook. It provides external system with uniform interface to identification of page of books and self-descriptive messages. example RESTful address is http://bukle.org/@PrideAndPrejudice/1/3/82. The meaning of address is the title of book is Pride and prejudice, the chapter is first chapter and the 3rd page of 82 page.

6. Conclusion and Future works

In this paper, we discussed collaborative social tagging for eBook using external DSL approach. We provides (1) DSL by which authors can write HTML5 rich contents ebook and tag resources, (2) tag model that makes users enhance book by tagging resources, (3) RESTful address style of ebook resources by which other system can identify self-descriptive resources of book.

To achieve the goal, we provide Bukle language by which author and users can author and enhance ebook. As a domain-specific language Bukle provides a simple yet expressive language for authoring and tagging books that would otherwise be more difficult to express with a general purpose language.

It is somewhat difficult for novice social tagger to tag books using textual language. Further work includes visual DSL approach and tools by using that the unskilled users could tag book easily. In order that future work also includes text-to-visual DSL transform engine.

UX research is also required to social tag and to author book with ease. To tackle the above questions we are looking at using visual notation which is focusing visual syntax.

References

[1] M. Hart, "Project Gutenberg," *Project Gutenberg*. [Online]. Available: http://www.gutenberg.org/. [Accessed: 18-Sep-2014].

[2] A. Mathes, "Folksonomies-cooperative classification and communication through shared metadata," *Comput. Mediat. Commun.*, vol. 47, no. 10, pp. 1–13, 2004.

[3] D. Ghosh, *DSLs in action*. Greenwich, Conn.: Manning, 2011.

[4] B. Kasdorf, "EPUB 3: Not Your Father's EPUB," *Inf. Stand. Q.*, pp. 4–11, 2011.

[5] "Timeline - Art Museum," *App Store*. [Online]. Available: https://itunes.apple.com/us/app/timeline-art-museum/id562792427?mt=8. [Accessed: 22-Sep-2014].

[6] "Timeline - U.S. History," *App Store*. [Online]. Available: https://itunes.apple.com/us/app/timeline-u.s.-history/id552887553?mt=8. [Accessed: 22-Sep-2014].

[7] K. Eaton, "Mobile Apps Bring History Alive," *The New York Times*, 06-Aug-2014.

[8] L. C. Vincent and et al, "Electronic Content Management and Delivery Platform."

[9] D. Daylamani Zad and H. Agius, "An MPEG-7 Profile for Collaborative Multimedia Annotation," in *The Handbook of MPEG Applications*, rios C. Angelides and H. Agius, Eds. John Wiley & Sons, Ltd, 2010, pp. 263–291.

[10] C.-C. Kiu and E. Tsui, "TaxoFolk: A hybrid taxonomy–folksonomy structure for knowledge classification and navigation," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 6049–6058, May 2011.

[11] A. Peleteiro, M. Rey-López, J. A. Martín-Pernas, J. C. Burguillo, and F. A. Mikic-Fonte, "TAGGEMENDOR: Relating multimedia using folksonomies," in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, 2011, pp. 221–222.

[12] F. P. Stappers, S. Weber, M. A. Reniers, S. Andova, and I. Nagy, "Formalizing a domain specific language using SOS: an industrial case study," in *Software*

Language Engineering, Springer, 2012, pp. 223–242.
[13] B. Langlois, C.-E. Jitia, and E. Jouenne, “DSL classification,” in *OOPSLA 7th Workshop on Domain Specific Modeling*, 2007.
[14] D. Kramer, T. Clark, and S. Oussena, “MobDSL: A Domain Specific Language for multiple mobile platform deployment,” in *Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on*, 2010, pp. 1–7.
[15] M. Funk and M. Rauterberg, “PULP scription: a DSL for mobile HTML5 game applications,” in *Entertainment Computing-ICEC 2012*, Springer, 2012, pp. 504–510.
[16] B. Heiko, C. Michael, E. Sven, E. Moritz, F. Peter, K. Jan, W. Knut, and Z. Sebastian, “Xtext User Guide.” Eclipse Foundation, 2010.
[17] E. Sven and Z. Sebastian, “Xtend,” 2011. [Online]. Available: http://www.eclipse.org/xtend/documentation.html.
[18] A. W. . Furtado, A. L. . Santos, and G. L. Ramalho, “Digital Games Development Automation through Domain-Specific Languages: an End-to-End Approach,” presented at the SBLP 2008, 2008.
[19] C. Paul and N. Linda, *Software product lines*. Pittsburgh: Software Engineering Institute, 2002.
[20] M. Riebisch, “Towards a more precise definition of feature models,” *Model. Var. Object-Oriented Prod. Lines*, pp. 64–76, 2003.
[21] E. M. Maximilien, A. Ranabahu, and K. Gomadam, “An online platform for web apis and service mashups,” *Internet Comput. IEEE*, vol. 12, no. 5, pp. 32–43, 2008.
[22] R. Walter and M. Masuch, “How to integrate domain-specific languages into the game development process,” in *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, 2011, p. 42.
[23] J. Webber, S. Parastatidis, and I. Robinson, *REST in practice*. Farnham; Sebastopol, Calif.: O’Reilly, 2010.
[24] S. Ovadia, “Markdown for Librarians and Academics,” *Behav. Soc. Sci. Libr.*, vol. 33, no. 2, pp. 120–124, 2014.