Wedata: A Wiki System for Service Oriented Tiny Code Sharing

Koichiro Eto
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba Central 2, 1-1-1
Umezono, Tsukuba, Ibaraki,
305-8568, JAPAN
k-eto@aist.go.jp

Masahiro Hamasaki
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba Central 2, 1-1-1
Umezono, Tsukuba, Ibaraki,
305-8568, JAPAN
hamasaki@ni.aist.go.jp

Hideaki Takeda
National Institute of Informatics (NII)
Hitotsubashi, 2-1-2,
Chiyoda-ku, Tokyo, 101-0003,
JAPAN
takeda@nii.ac.jp

WikiSym '12, Aug 27—29, 2012, Linz, Austria.
ACM 978-1-4503-1605-7/12/08.

Abstract
A new trend for applications for the Internet is to create and share tiny codes for applications like site-specific codes for a broswer extension. It needs a new tools to share and distribute such codes efficiently. We built a Wiki site called Wedata which stores tiny code for a particular service. Wedata has three features: machine readability, code sharing, and service orientation. Many developers already use Wedata for browser extensions. More than 1,300,000 users are using Wedata. As described in this paper, we describe the Wedata system, usage statistics, and the behavior of open collaboration on the system.

Author Keywords
Wiki; Metadata; Open source; Open Collaboration

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

General Terms
Design, Experimentation

Figure 1: Screenshot of Wedata (`http://wedata.net/`).

Introduction

When creating a program to process several web sites on the internet, one must write different code for each web site. For example, creation of a scraper program to extract body text from several web sites requires specification of the location of the body text of the web pages. One can usually specify the body text using a regular expression or XPath. We designate that small code as "tiny code".

In this study, we build Wedata as a Wiki system for sharing tiny code. Wedata has three features: machine readability, code sharing, and service orientation. Conventional Wiki systems are unsuitable for machine reading and writing. Using Wedata, one can specify a scheme for each dataset, and can add items along with the scheme. Furthermore, one can obtain the entire dataset using JSON or XML format easily.

The Wedata system is used by various applications, mainly by browser extensions. For example, AutoPagerize is a browser extension that adds the next page onto the current page only by scrolling the page. In AutoPagerize, the extension requires a dataset for the position of the "next" link and the body text of the web pages. AutoPagerize stores the dataset on Wedata.

Datasets on Wedata are designed for each particular application. Therefore, we designate these datasets as service-oriented. The conventional means used to provide a dataset are to distribute the dataset with the application (browser extension) itself. However, we chose to store and share the dataset on a Wiki system.

In the open source software (OSS) development model, core developers check patches contributed by various users. If the patches are OK, then they commit them. However, one can edit or store the code at any time on Wedata, and the code is distributed without review. Therefore, the application developer must design the dataset scheme carefully to avoid including or creating security issues.

Related works

There are some Wiki systems for machine readable data, such as Google DataWiki[1], SemanticMediaWiki[6], and OntoWiki[3]. These systems are intended to share data, not code. ScraperWiki[2] is a service for sharing codes of scraper, which extracts data from Web pages. ScraperWiki and Wedata have the same purpose, but their approaches differ. ScraperWiki shares code with respect to each Web site, which is a target of scraping. It is a content-oriented approach. Wedata shares codes with respect to each service that uses extracted data from some Web sites. It is a service-oriented approach.

CoScripter[5] is a system for sharing codes that automate processes in a web browser. However, code types are different. In CoScripter, one service requires one code. In Wedata, one service, such as AutoPagerize, requires many tiny codes. It means that a creator of a service (e.g., AutoPagerize) are more likely to have a motivation of sharing codes using Wiki approach because it requires "many" codes. Furthermore, users are more likely to have a motivation to uploads codes because it requires "tiny" codes which users can generate easily.

Wedata

Wedata is a Wiki system for sharing tiny code. When creating an account on Wedata with OpenID, one can launch a new dataset on Wedata. Each dataset on Wedata has a schema. Users can register or edit items to the dataset according to the schema. Users can download the items in JSON form. Therefore, one can easily use the

Attribute	Value
url	`\^https?://[\^./]+\.google(?:\.[\^./]{2,3}){1,2}/(?:c(?:se|ustom)|search|webhp|\#)`
nextLink	`id("pnnext")|id("navbar navcnt nav")//td[span]/following-sibling::td[1]/a|id("nn")/parent::a`
pageElement	`id("res")[not(@role)]/div[ol or div] | id("ofr") | id("rso")`
exampleUrl	https://www.google.com/search?q=AutoPagerize

Table 1: Example of item on AutoPagerize dataset.

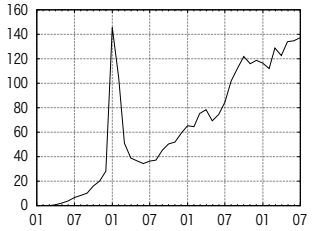

Figure 2: The growth of monthly unique IPs on Wedata.

dataset from an application. Table 1 is an example of an item on Wedata that AutoPagerize uses. The "url" is the regular expression of URL for the corresponding Web page. XPath "nextLink" shows the link of the following page. XPath "pageElement" shows the body of the web page.

Empirical results

In this section, we present results of analysis of users' logs in Wedata. The service started at March, 2008. As of June 2011, it has more than 47 million page views per month and 1.3 million unique IPs are accessed. Figure 2 shows the growth of unique IPs. Most of users access Wedata via application.

Table 2 shows the top 5 popular datasets in Wedata. "UIP" represents the number of unique IPs which accessed the dataset. "Editors" shows the number of users who registered or edited data. "Items" shows the number of data. The number shown in parenthesis is a rate of items edited by multiple editors (we call this type of items as C-Items). Editing with some editors is one characteristic of Wiki. If this value is high, then that dataset is suitable for use with a Wiki approach. Figure 3 shows the growth of AutoPagerize dataset.

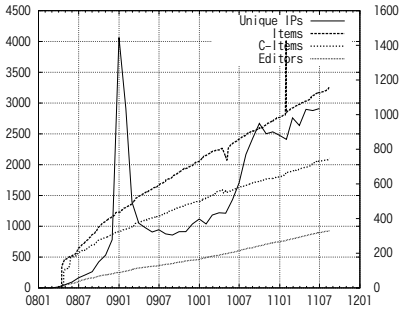

Figure 3: The growth of AutoPagerize dataset.

Implementation of Wedata is a key value open database platform. So it dose not restrict a type of data to tiny code. However, most of the popular datasets share tiny codes. In Table 2, all datasets with the exception of Favicons shared tiny codes. Favicons is a dataset to share URL of favicon.

Dataset	UIP	Editors	Items
AutoPagerize	1035312	927	3251 (0.64)
LDRize	327431	61	515 (0.45)
HatenaBookmarkUsersCount	325698	6	8 (50.0)
LDRFullFeed	33903	353	3085 (0.27)
Favicons	23946	3	7059 (0.00)

Table 2: Top 5 popular datasets in Wedata.

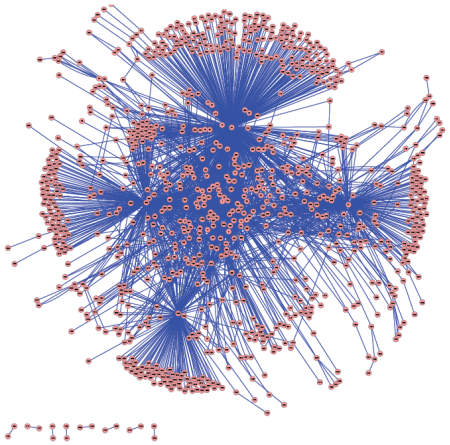

Figure 4: Sequential collaboration network on Wedata.

Figure 4 shows a sequential collaboration network[4] in Wedata. This network is generated from anteroposterior relationships of editing among editors. Wedata has 155 datasets, but the sequential collaboration network has one large cluster, which means that editors can edit data across datasets.

Discussion

In Wedata, a user establishes a dataset for use from specified services. However, some datasets are used by other services. For example, the AutoPagerize dataset is used by AutoPatchWork, which is a web browser extension with a similar function. The LDRize dataset is used by the Hatena bookmark extension. It is a web browser extension for Hatena Bookmark, which is the most famous social bookmarking service in Japan. The Hatena bookmark extension uses not only a LDRize dataset but also own dataset to cover data which cannot be followed by LDRize. AutoPathWork and Hatena bookmark extension might be free riders. However, they bring new users who might become editors. Our approach provides open collaboration of two types, open collaboration among people on Wedata as a platform of co-editing tiny code, and open collaboration among services on Wedata as a platform for sharing dataset of tiny codes.

Conclusion

As described in this paper, we presented an overview and operation of the Wedata system, which enables sharing of service-oriented tiny code. Many applications and browser extensions have adopted the "service oriented tiny code sharing by Wiki" method provided by Wedata system. Many users use them, we think that the method is useful.

References

[1] Google DataWiki, 2010. `http://datawiki.googlelabs.com/`

[2] Scraper Wiki, 2010. `https://scraperwiki.com/`

[3] S. Auer, S. Dietzold, T. Riechert, and T. Riechert. Ontowiki - a tool for social, semantic collaboration. In *Proc. of ISWC '06*, pages 736–748, 2006.

[4] T. Iba and S. Itoh. Sequential collaboration network of open collaboration. In *International Workshop and Conference on Network Science '09*, 2009.

[5] G. Leshed, E. M. Haber, T. Matthews, and T. Lau. Coscripter: Automating & sharing how-to knowledge in the enterprise. In *CHI'08*, pages 1719–1728, 2008.

[6] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic wikipedia. In *Proc. of WWW '06*, pages 585–594, 2006.