

ContextGrid: A contextual mashup-based collaborative browsing system

Jason J. Jung

Published online: 26 July 2011
© Springer Science+Business Media, LLC 2011

Abstract Due to a large amount of resources (i.e., information and knowledge) available on world wide web, it has been more difficult for users to effectively find relevant web resources. Most of the current web browsing methods and systems have been investigated to apply adaptive approaches which can extract personal contexts (e.g., interests and preferences) of the users. In this paper, we propose a contextual mashup-based collaborative browsing (co-browsing) platform, called *ContextGrid*, for providing online users with various knowledge sharing services. Particularly, the proposed mashup scheme can integrate heterogeneous pieces of information collected by various Open APIs, and assist the users to decide which partners should be selected for mutual collaborations. In order to evaluate the proposed mashup-based method, we have implemented a co-browsing platform which can exchange bookmarks, and measured whether the contextual mashup scheme makes a meaningful influence on improving the performance of the co-browsing process with multiple users.

Keywords ContextGrid · Collaborative browsing · Information searching · Knowledge sharing · Open API · Contextual mashup

1 Introduction

Information searching tasks on the web has been regarded as an important challenge on many research areas and communities, e.g., search engines and web-based information systems. There have been many issues to improve the performance of such information searching tasks on the web. For example, many studies on web crawlers have been trying to deal with indexing large-scale web resources which have been linked with each other. In addition, there are some studies to rank the web resources retrieved from a certain query. Thus, we can note that in the context of this information searching tasks, the problematic characteristics of the web environment are (i) a large amount of the information (so called “information overload” problem), and (ii) dynamics of the information on the web.

The main issue that we focus on in this paper is *user navigation* on the web. This navigation process is a series of interactions between a user and web for finding relevant information (Dömel 1995; Jung 2005). There have been a number of systems exploiting user profiling methods through analyzing explicit and implicit behaviors of each user (e.g., query logs, visiting frequencies, and so on). For example, the personal assistant agent system (Amant et al. 1998) is able to predict next actions of the users, thereby enabling it to perform such actions as proactively prefetching and showing candidate web pages based on the user’s preferences (Lieberman et al. 2001).

However, due to such problematic characteristics of the web, navigation on the web is one of the most lonely and time-consuming tasks (Maes 1994). In contrast to those single user-centered approaches, in this study, we propose an alternative navigation way of supporting

J. J. Jung (✉)
Knowledge Engineering Laboratory,
Department of Computer Engineering,
Yeungnam University, 214-1 Dae-Dong,
Gyeongsan, 712-749 Republic of Korea
e-mail: Ontology.Society@gmail.com

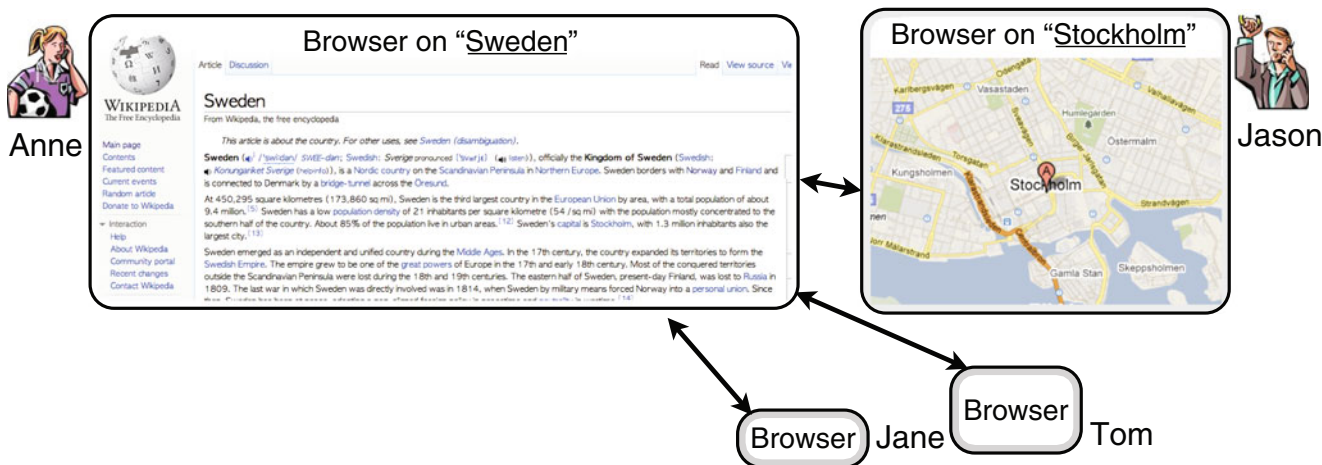


Fig. 1 Co-browsing among multiple users

collaborations among multiple users in real time. For example, in Fig. 1, Anne can get the relevant information from Jason, because their browsers can detect that they are in the similar context (e.g., “Sweden” and “Stockholm”). Collaborative browsing (hereinafter, co-browsing) proposed in this paper is an approach by which users can exchange knowledge and resources with their like-minded neighbors while searching for information on the web. By online communication between users, they can acquire not only relevant resources, but also many types of user experiences (e.g., heuristics and know-how) from the others, such as how to select and rank the search results, how to make an appropriate sequence of queries, and how to choose the best searching method, as well as providing the other users with their own knowledge. Hence, we expect that improve the performance of information searching on the web can be improved.

To do so, we design a *ContextGrid* model to compare user contexts and to integrate them for supporting contextual collaborations between users. In the co-browsing system, we have to consider how to recognize current contexts of each user, and, more importantly, how to compare the contexts between two users. These processes are also important for implicitly conducting various tasks, e.g., generating queries, filtering the query results, and recommending relevant results. Particularly, in this study, we focus on analyzing a set of user bookmarks for extracting user interests. The co-browsing system can easily be aware of the current user contexts by matching the current page view to these user interests. Also, for better understanding on recommended information, open API-based mashups are exploited to enrich the recommendations with additional information from various information sources. For example, in Fig. 1, even though Anne has been

received relevant information from Jason, it is difficult for her to understand it correctly and efficiently. The system may be able to find out that Jason is in a similar context from his browser illustrating the geographical location of “Stockholm” by using Google map API.¹

The outline of this paper is as follows. In the following Section 2, we explain background knowledge about co-browsing systems and mashup techniques. Also, several previous studies related to this paper are presented. Section 3 describes how to extract user interests and contexts for co-browsing, and how to enrich the information by using the existing mashups. In Section 4, the architecture of the proposed co-browsing system is presented. After the evaluation of the proposed system is discussed in Section 5, we draw a conclusion of this paper in Section 6.

2 Background and related work

In this section, we explain two main technologies composing the proposed co-browsing system, which are (i) co-browsing, and (ii) mashups. Furthermore, the related systems on these technologies are compared to the system.

2.1 Collaborative browsing

Since Rodden (1991) has classified co-browsing systems into four classes,² there have been a number of systems for supporting collaborations between people during

¹<http://code.google.com/apis/maps/index.html>

²In terms of temporal and spatial characteristics, each co-browsing system can be either synchronous or asynchronous, and either local or remote.

web browsing. As some traditional co-browsing systems, ARIADNE (Twidale and Nichols 1998), Web-Watcher (Joachims et al. 1997), and Let’s Browse (Lieberman et al. 1999) have shown some interesting features. In Let’s Browse, people have to use the infrared sensors for the purpose of detecting the presence of users without any explicit actions, and people can instantly exchange information between them. ARIADNE system has to record all of the searching processes in a digital library (Twidale and Nichols 1998), thus allowing other users this information to be visualized and reused whenever they want.

However, the most significant difference between such co-browsing systems is how to extract user preferences from personal activities. While Let’s Browse and ARIADNE use statistical scheme (e.g., TF-IDF term weighting) to analyze the keyword frequency of the web pages visited by the corresponding users, WebWatcher and BISAgent (Jung 2005) focus on incremental learning approaches from the aggregated user activities. More exactly, BISAgent deals with the extraction of the user contexts through the semantic learning of their activities from a set of bookmarks. Ontology learning process can be conducted by information collected from heterogeneous sources, so that the semantic structure can be retrieved and applied to document management and clustering. Similarly, Magpie (Domingue et al. 2004) has tried to employ semantic web technologies to support co-browsing between users.

2.2 Mashup by open APIs

The mashup is a web application (or a web page) that uses and combines data, presentation or functionalities from two or more sources to create new services. There are various technologies and standards for conducting data presentation and user interactions. Such technologies are HTML/XHTML, CSS, Javascript, Asynchronous Javascript and XML (Ajax). Especially, in case of Web services-based functionality, we can employ open API services, which are based on XMLHttpRequest (XHR), XML-RPC, JSON-RPC, SOAP and REST, to improve our own web applications.

Recently, a mashup-based web development has been more widely applied in many domains, e.g., healthcare (Cheung et al. 2008; Belleau et al. 2008), geospatial domain (Becker and Bizer 2009; De Longueville 2010), and education (Mason and Rennie 2007). According to ProgrammableWeb,³ which is a

directory service of mashup applications, about 60% of the current mashups are related to geographical mapping services.

In the area of web browsing application, particularly, rich web applications (RWA) have been developed to call available APIs for collecting various types of information from different sources. They mainly expect that the information might be helpful for the users’ searching tasks on the web.

3 Contextual collaboration

Co-browsing system needs to find out what the corresponding users are currently doing. The user context is the most important information for making relevant connections between users (Koo et al. 2011). In fact, there can be various possibilities to be aware of user contexts from both explicit and implicit use activities. In this study, we focus on only two activities for the user context awareness process. We assume that (i) a set of bookmarks collected by a user and (ii) current web page are related to the user context.

3.1 Contexts from user browsing

In this paper, the current contexts during user browsing can be obtained by matching between the current webpage and a set of bookmarks. This matching process is to find out semantic relationships between the current webpage and a set of bookmarks (Jung 2011a). It means that the system can determine which bookmarks are most semantically related to the current webpage through this matching process. Thereby, personal ontology for each user should be constructed, before the matching.

Definition 1 (Personal ontology (Jung 2008a)) For a given set of annotations $\mathcal{A}_{\mathcal{I}}$ from a set of bookmarks \mathcal{I} , a personal ontology \mathbb{O} is represented as

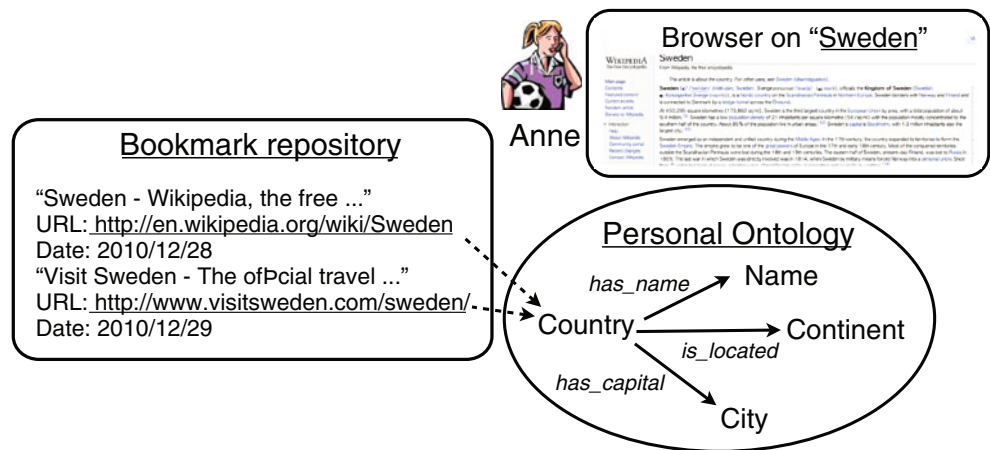
$$\mathbb{O} := \langle \mathcal{C}, \mathcal{R}, \mathcal{I}, \mathcal{E}_{\mathcal{R}}, \mathcal{A}_{\mathcal{I}} \rangle \tag{1}$$

where \mathcal{C} , \mathcal{R} and \mathcal{I} are a set of classes and a set of relations (e.g., equivalence, subsumption, and disjunction) between the classes, respectively. $\mathcal{E}_{\mathcal{R}} \subseteq \mathcal{C} \times \mathcal{C}$ is a set of relationships between classes, represented as a set of triples $\{\langle c_i, r, c_j \rangle | c_i, c_j \in \mathcal{C}, r \in \mathcal{R}\}$. Also, $\mathcal{A}_{\mathcal{I}}$ indicates a set of annotations between a bookmark in \mathcal{I} and a class in \mathcal{C} .

However, in practice, it is difficult to automatically build the personal ontologies from a set of bookmarks. Thus, during bookmarking the webpages into a local

³<http://www.programmableweb.com/>

Fig. 2 Personal ontology from bookmarks. The dotted arrows indicate the semantic annotation of the bookmarks



repository, as shown in Fig. 2, users are asked to create their own personal ontologies, and also to annotate the bookmarks by using the ontologies. In this study, we represent the personal ontology by using OWL2.⁴

A personal ontology is extended with subjective descriptions about contents in personal bookmarks. As the users bookmark more webpages, their personal ontologies are incrementally enlarged by adding new classes, merging ontology fragments, and declaring additional matches between classes. User context is represented by using his own personal ontology, while browsing for searching for a certain resource. Note that because the semantic metaphor of the resource might be uncertain and ambiguous, each browsing system is trying to select the most relevant context among all possible candidates and to maximize understandability of neighbors' systems. We have extended ontology-based context representation (Jung 2008a) in terms of web browsing. The context as a set of concepts can be obtained by a matching function between the personal ontology and the webpage at a certain moment.

Definition 2 (Context) When a user u_k visits a webpage w_i at time t , his context $ctx_k^{(t)}$ obtained by a matching function \mathcal{M} is given by

$$ctx_k^{(t)} = \{c_i | c_i \in \mathcal{M}(\mathbb{O}_k^P, res^{(t)})\} \tag{2}$$

where $res^{(t)}$ is a set of terms which are highly occurred in webpage w . Also, the matching function \mathcal{M} will be discussed later.

Additionally, the set of terms $res^{(t)}$ from the current webpage can be simply selected if the number of occur-

rences of a term is more than a certain threshold τ . This threshold will be decided by empirical evaluation.

3.2 Finding neighbors

Once the user contexts are extracted from the user browsing activities, the system has to find only the collaborators who can exchange relevant information with him. It is also based on matching between two user contexts in a same moment. Thus, given two users u_i and u_j at time t , the matching (Jung 2008b, 2010a, b) can be simply represented as

$$\begin{aligned} \mathcal{M}(ctx_i^{(t)}, ctx_j^{(t)}) &= \sum_{E \in \mathcal{N}(C)} \pi_E^C MSim_Y(E(ctx_i^{(t)}), E(ctx_j^{(t)})) \end{aligned} \tag{3}$$

where $\mathcal{N}(C) \subseteq \{E^1 \dots E^n\}$ is the set of all relationships in which classes can be involved (for example, attributes, subclasses, or instances). The weights π_E^C should be normalized (i.e., $\sum_{E \in \mathcal{N}(C)} \pi_E^C = 1$). Since we restrict the user context in Eq. 2 to class labels (L) and three relationships in $\mathcal{N}(C)$, which are the superclass (E^{sup}), the subclass (E^{sub}) and the sibling class (E^{sib}), Eq. 3 is rewritten as:

$$\begin{aligned} \mathcal{M}(ctx_i^{(t)}, ctx_j^{(t)}) &= \pi_L^C sim_L(L(ctx_i^{(t)}), L(ctx_j^{(t)})) \\ &+ \pi_{sup}^C MSim_C(E^{sup}(ctx_i^{(t)}), E^{sup}(ctx_j^{(t)})) \\ &+ \pi_{sub}^C MSim_C(E^{sub}(ctx_i^{(t)}), E^{sub}(ctx_j^{(t)})) \\ &+ \pi_{sib}^C MSim_C(E^{sib}(ctx_i^{(t)}), E^{sib}(ctx_j^{(t)})). \end{aligned} \tag{4}$$

⁴<http://www.w3.org/TR/owl2-overview/>

where the set functions $MSim_C$ compute the similarity of two entity collections. Consequently, the matching between two user contexts returns a value in $[0, 1]$. If it is 1, we assume that they are working exactly same tasks.

As a matter of fact, a similarity between two sets of classes in the user contexts can be established by finding a maximal matching maximizing the summed similarity between both classes:

$$MSim_C(S, S') = \frac{\max(\sum_{(c,c') \in Pairing(S,S')} (Sim_C(c, c'))}{\max(|S|, |S'|)}, \tag{5}$$

in which $Pairing$ provides any possible matching between these two sets of classes. Finally, the context matching process can find a number of neighbors who are working on the semantically related tasks in real time.

To implement this, in practice, we have exploited a multi-agent based platform where each agent is embedded into each browser of a user.

As shown in Fig. 3, all of the agents have to send the user contexts to facilitator agent which is located in the middle. The facilitator can play an important

bridge role among the other agents, once it conduct the context matching process between all possible pairs of users. For example, the facilitator can easily find that ‘Anne’, ‘Tom’ and ‘Hong’ are looking for information about “Sweden”. By the facilitator’s suggestion, these two corresponding agents (or browsers) will open the channel for sharing their views.

3.3 Information integration by semantic mashup

The next issue is to integrate various types of web data (e.g., simple textual documents, images, maps, multimedia data, and so on) collected from open APIs. For instance, in Fig. 3, even though the three users ‘Anne’, ‘Tom’ and ‘Hong’ are working similar tasks, it is difficult to detect the user contexts and also more difficult to directly compare them.

To overcome the problem, we have proposed a “semantic mashup” scheme for integrating heterogeneous information from multiple sources. This scheme is based on a set of adapters. The adapter can transform the semantics into the other one which is universally understandable. Thus, the facilitator can efficiently enrich the information by such integration tasks by open APIs. For example, the adapter for a mapping service should

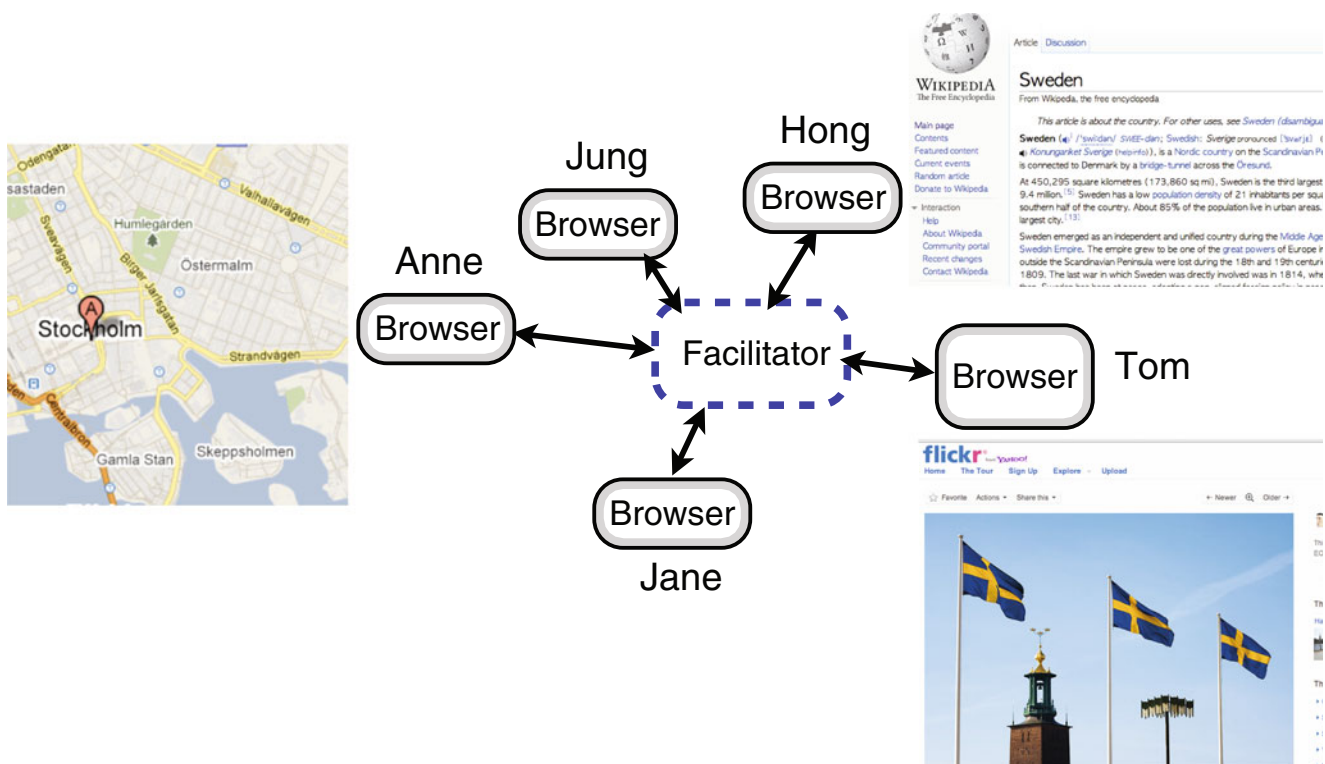


Fig. 3 A multi-agent based architecture for co-browsing

Table 1 Targeted APIs for adapters

APIs	Inputs	Outputs	Examples
Mappings	Geographical coordination	City names	Google map API
Photos	Tagged pictures	Tag set	Flickr API
Bookmarks	Tagged bookmarks	Bookmark set	Delicio.us API

be in advance selected for transforming a geographical location (e.g., longitude and latitude) into a city (or a country). Then, the city's name can be compared to the label of the class in the personal ontologies for conducting the context matching process.

Thus, in this study, we have been focusing on three main types of open APIs, as shown in Table 1. The common task of these adapters is to transform the output of the APIs into a textual form for making them to be comparable.

4 Mashup-based co-browsing system

In this section, we want to describe the whole architecture of the proposed mashup-based co-browsing system.

As shown in Fig. 4, the system consists of two main parts which are (i) client-side component and (ii) server-side one. Each client needs to employ the personal agent component which has five modules.

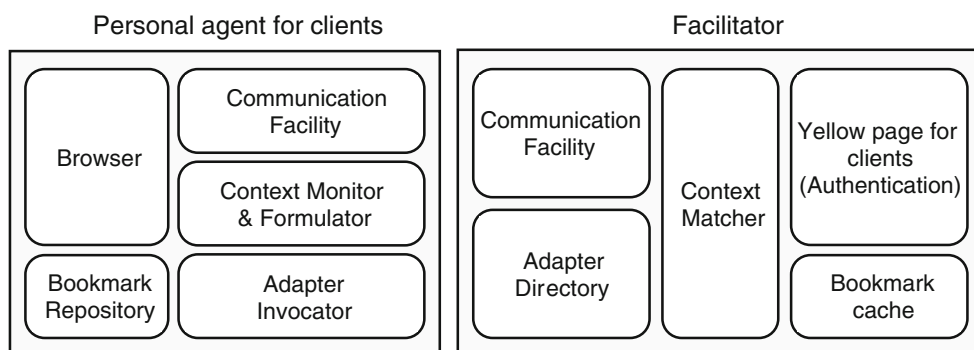
- Bookmark repository. This is a simple local directory for the bookmark files saved by the user.
- Browser. Similar to the ordinary web browsers, users can take any activities through this module. It means the users are interacting with the personal agent. For illustrating a number of additional views retrieved from other neighbors, it has employed a tab component.

- Communication facility. It is indicating all gadgets related to communications. Mainly, in client side, it has a function for generating client socket for the communication.
- Context monitor and formulator. This module can aggregate all of the user activities (Jung 2009) during browsing. More importantly, it has two sub-modules for formulating the user context.
 - Ontology editor: An important characteristic is that the ontology can import other ontologies and a set of partitioned ontology fragments. For example, some standard (upper-level) ontology can be employed with OWL vocabularies like `owl:imports`.
 - Semantic annotator: The monitored user activities are matched to the personal ontologies. Also, the saved bookmarks are annotated with semantics from the ontology.
- Adapter invocator. It is to enrich the information with the resources retrieved by open APIs. Thus, this can select proper adapter to transform.

As the second part, the service-side component is composed of five modules.

- Communication facility. It indicates communication features. Especially, server side needs to manage multiple server, simultaneously.
- Adapter directory. It is a list of adapter. Personal agents from clients can look up this directory to receive the proper adapter for transformation.
- Yellow page for clients. It is a list of users who want to access this system. Simple authentication process is conducted by this module.
- Context matcher. It can collect the user contexts from all clients, and try to understand the relationships between the contexts by similarity-based matching.

Fig. 4 The system architecture of mashup-based co-browsing system



- **Bookmark cache.** It is a simple cache memory for some bookmarks which are frequently used. It is important for improve the scalability of the system with a large number of clients.

To implement this multi-agent based co-browsing platform, BISAgent (Jung 2005) has been extended to exchange more relevant resources between users with a number of different open APIs.

5 Evaluation

The proposed co-browsing system has been evaluated by analyzing user feedbacks. We have invited 68 undergraduat students to prove the performance of the proposed system. In order to reduce the scope of their tasks, we asked them to select one questions from ten questions related to “vacation.”, and to answer the question. Thus, they had to seek information related to their contexts.

We have focused on two evaluation issues, which are (i) performance of information searching, and (ii) user satisfaction of information searching.

5.1 Performance of information searching

There were two user groups which are G_{co} (34 students) and G_{nco} (34 students). While G_{co} are allowed to use the proposed co-browsing system for collaboration, G_{nco} was based on single browsing. Surely, students in G_{nco} were used other communication tools, e.g.,

email, instant messengers, and so on. During two weeks (September 6th to 19th, 2010), they had to collect as many relevant webpages as possible. As shown in Fig. 5, we have found that G_{co} with the co-browsing system shows significantly higher performance than G_{nco} .

In the initial stage (about 3 days), they were quite similar results. We have realized that the students in G_{co} needed to learn how to use the ontology editor and create their personal ontologies. Once they were good at such tools, the performance of information searching and sharing has been quickly increased.

5.2 User satisfaction

As second evaluation issue, we have asked the students how much they were satisfied with the results which were recommended by the system, in terms of two indicators, (i) recommendation speed and (ii) information enrichment by open APIs.

As shown in Table 2, most of them were satisfied with those two indicators with 82.4% and 94.1%, respectively. After we have asked why two students were not satisfied with the first indicator, we realized that the system bothered the students to build their own personal ontologies, and they felt too difficult on the tasks.

From this user survey, we found that open APIs are the important sources to enrich the simple textual documents (e.g., webpages). Even though we have focused on only the three open APIs, the students have shown very high rate of satisfaction.

Fig. 5 Performance of information searching by two user groups G_{co} and G_{nco}

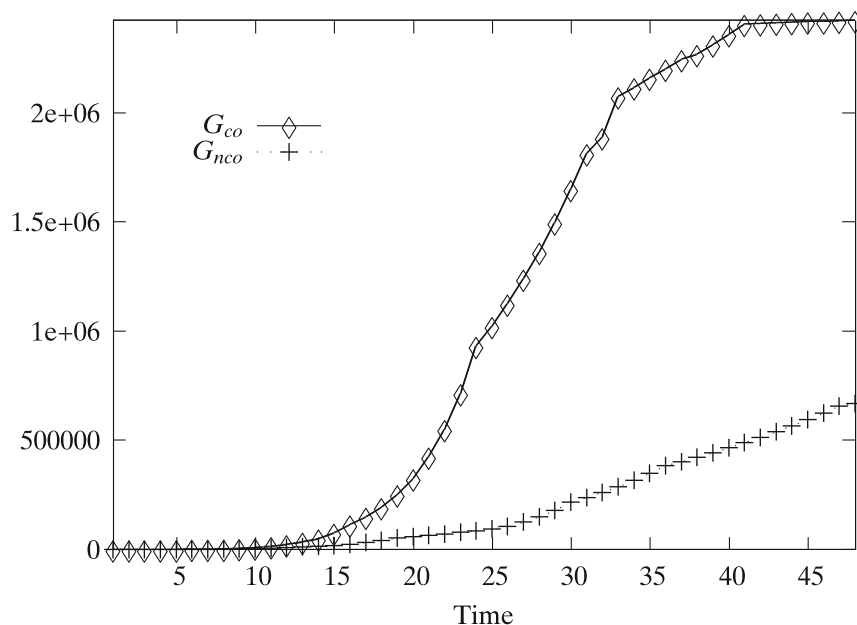


Table 2 User satisfaction

	Recommendation speed	Information enrichment by open APIs
Vary satisfied	16	11
Satisfied	12	21
Normal	4	2
Unsatisfied	2	–
Very unsatisfied	–	–

6 Concluding remarks and future work

In this paper, we have proposed and evaluated a mahup-based co-browsing systems by using open APIs. To do so, we have defined a novel user context formalization and representation of web browsing processes by deriving relevant semantics from ontologies. Thus, a software agent can automatically realize and understand other users' contexts by communicating with their agents.

As a conclusion of this work, we have shown that the proposed co-browsing system can efficiently support collaborations among users during searching for information on the web. Particularly, open API have been playing an important role of enriching the information with mapping and photos.

In the future, we are planning to extend the proposed system to various service-oriented domains (Jung 2011b). The first case will be on e-learning area. This area has been regarded as an important problem on blended learning.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2011-0017156).

References

- Amant, R. St., Long, T., & Dulberg, M. S. (1998). Experimental evaluation of intelligent assistance for navigation. *Knowledge-Based Systems*, 11(1), 61–70.
- Becker, C., & Bizer, C. (2009). Exploring the geospatial semantic web with dbpedia mobile. *Journal of Web Semantics*, 7(4), 278–286.
- Belleau, F., Nolin, M.-A., Tourigny, N., Rigault, P., & Morissette, J. (2008). Bio2rdf: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, 41(5), 706–716.
- Cheung, K.-H., Yip, K. Y., Townsend, J. P., & Scotch, M. (2008). Hcls 2.0/3.0: Health care and life sciences data mashup using web 2.0/3.0. *Journal of Biomedical Informatics*, 41(5), 694–705.
- De Longueville, B. (2010). Community-based geoportals: The next generation? concepts and methods for the geospatial web 2.0. *Computers, Environment and Urban Systems*, 34(4), 299–308.
- Dömel, P. (1995). Webmap: A graphical hypertext navigation tool. *Computer Networks and ISDN Systems*, 28(1–2), 85–97.
- Domingue, J., Dzbor, M., & Motta, E. (2004). Collaborative semantic web browsing with magpie. In C. Bussler, J. Davies, D. Fensel, & R. Studer (Eds.), *Proceedings of the 1st European semantic web symposium (ESWS 2004), May 10–12. Lecture Notes in Computer Science* (Vol. 3053, pp. 388–401). Heraklion, Crete, Greece: Springer.
- Joachims, T., Freitag, D., & Mitchell, T. (1997). Webwatcher: A tour guild for the world wide web. In *Proceedings of the 15th international joint conference on artificial intelligence* (pp. 770–775).
- Jung, J. J. (2005). Collaborative web browsing based on semantic extraction of user interests with bookmarks. *Journal of Universal Computer Science*, 11(2), 213–228.
- Jung, J. J. (2008a). Ontology-based context synchronization for ad-hoc social collaborations. *Knowledge-Based Systems*, 21(7), 573–580.
- Jung, J. J. (2008b). Query transformation based on semantic centrality in semantic social network. *Journal of Universal Computer Science*, 14(7), 1031–1047.
- Jung, J. J. (2009). Knowledge distribution via shared context between blog-based knowledge management systems: A case study of collaborative tagging. *Expert Systems with Applications*, 36(7), 10627–10633.
- Jung, J. J. (2010a). An empirical study on optimizing query transformation on semantic peer-to-peer networks. *Journal of Intelligent & Fuzzy Systems*, 21(3), 187–195.
- Jung, J. J. (2010b). Reusing ontology mappings for query segmentation and routing in semantic peer-to-peer environment. *Information Sciences*, 180(17), 3248–3257.
- Jung, J. J. (2011a). Service chain-based business alliance formation in service-oriented architecture. *Expert Systems with Applications*, 38(3), 2206–2211.
- Jung, J. J. (2011b). Boosting social collaborations based on contextual synchronization: An empirical study. *Expert Systems with Applications*, 38(5), 4809–4815.
- Koo, C., Wati, Y., & Jung, J. J. (2011). Examination of how social aspects moderate the relationship between task characteristics and usage of Social Communications Technologies (SCTs) in organizations. *International Journal of Information Management*, 31(5), 445–459.
- Lieberman, H., van Dyke, N., & Vivacqua, A. (1999). Let's browse: A collaborative browsing agent. *Knowledge-Based Systems*, 12(8), 427–431.
- Lieberman, H., Nardi, B. A., & Wright, D. J. (2001). Training agents to recognize text by example. *Autonomous Agents and Multi-Agent Systems*, 4(1–2), 79–92.
- Maes, P. (1994). Agents that reduce work and information overload. *Communications of ACM*, 37(7), 31–40.
- Mason, R., & Rennie, F. (2007). Using web 2.0 for learning in the community. *The Internet and Higher Education*, 10(3), 196–203.
- Rodden, T. (1991). A survey of cscw systems. *Interacting with Computers*, 3(3), 319–354.
- Twidale, M., & Nichols, D. (1998). Computer supported cooperative work in information search and retrieval. *Annual Review of Information Science and Technology*, 33, 259–319.

Jason J. Jung is an assistant professor of Computer Engineering Department at Yeungnam University, Korea. He was a postdoctoral researcher in INRIA Rhone-Alpes, France in 2006, and a visiting scientist in Fraunhofer Institute (FIRST) in Berlin, Germany in 2004. He received the B.Eng. in Computer Science and Mechanical Engineering from Inha University in 1999. He received M.S. and Ph.D. degrees in Computer and Information Engineering from Inha University in 2002 and 2005, respectively. His research topics are knowledge engineering on social networks by using machine learning, semantic web mining,

and ambient intelligence. He has about 30 international journal articles published in Knowledge-Based Systems, Information Retrieval, Information Processing & Management, Knowledge and Information Systems, and Expert Systems with Applications. Also, he is an editorial member of Journal of Universal Computer Science and International Journal of Intelligent Information and Database Systems. Moreover, he has been editing 10 special issues in Information Sciences, Journal of Network and Computer Applications, Computing and Informatics and so on.