

Shizuku2.0: Cooperative Reading Support System

MAO TSUNEKAWA, tsunekaw@slis.tsukuba.ac.jp
HARUKI ONO, milky@slis.tsukuba.ac.jp
KYOJI KONISHI, s0711516@u.tsukuba.ac.jp
KEITA TSUJI, keita@slis.tsukuba.ac.jp
ATSUSHI MATSUMURA, matsumur@slis.tsukuba.ac.jp
NORHIKO UDA, uda@slis.tsukuba.ac.jp
Graduate School of Library, Information and Media Studies, University of Tsukuba

ABSTRACT

The paper aims to propose cooperative reading, which is a reading support technique that allows library users to help each other. To achieve cooperative reading, it is necessary for a user to discover others with similar interests. Therefore, this paper also aims to develop and evaluate a recommendation function that recommends similar users using Nippon Decimal Classification (NDC) Tree Profiling. Is the user recommendation using NDC Tree Profiling effective in finding similar users? Which parameter of NDC Tree Profiling method is the most effective expression of users' interests? We developed the Shizuku2.0 system to support the creation of a library user community in which users help each other efficiently and mutually. We also designed and developed NDC Tree Profiling, which enables the creation of library user profiles, for the purposes of the user recommendation mechanism. To verify the effect of the user recommendation mechanism, we performed an experiment with 37 student users to calculate recall and precision. We found that the recommendation using NDC Tree Profiling is more effective than a random recommendation. However, we also recognized that there is room for improvement relative to a past information recommendation technique. Moreover, we found the second level of the NDC code could be the most effective expression of users' interests. In the discussion of the optimization of parameters, we propose a new way of implementing the NDC Tree, based on the second division of NDC, which is expected to improve creation of user profiles.

Keywords: Library 2.0; Impact of ICT on libraries; Cooperative reading; Digital library system; e-learning; Shizuku2.0

INTRODUCTION

Learning Commons and Library 2.0

The relationship between the library and the community has a great influence on the design of the library automation system. The recent movement toward Learning Commons and Library 2.0 has increased the importance of the community in the library.

The Learning Commons is an idea pertaining to the learning environment, advocated in many university libraries. In a Learning Commons, students or researchers, etc., can freely study and enter discussion with one other. The role of the university library lies in providing information to the library users, though recently, some university libraries have begun to offer places of study that take the form of a Learning Commons. Nagata (2008) states the most essential difference between libraries in the past and Learning Commons is the point of view about knowledge offered library users. The former assume that libraries possess extensive prior knowledge, whereas the latter assume that the knowledge is created through collaboration between library users. In a Learning Commons, a conventional OPAC and integrated library system (ILS) cannot meet the needs of library users because they can only support management and searching through resources such as books, magazines, and papers. It is necessary that a library automation system be in place to support the user in drawing on existing material and in forming new knowledge.

Library 2.0 is a movement that tries to incorporate elements of social media in an OPAC. This movement began to emerge following 2007 (Murakami, 2007). For instance, the Ann Arbor District Library in Michigan, United States, developed an OPAC that incorporates a social network service (SNS), called SOPAC (Blyberg, 2007). In Japan, Kyushu University provided a community function on their OPAC, in cooperation with an existing local community SNS (NDL,

2007). Brangier (2009) assumed that Digital Libraries are necessary to direct library users to discussion groups on appropriate topics.

For the reason stated above, it is necessary that library engineers consider the community of library users when they develop a new library automation system. In this paper, we purpose the design of a library automation system supporting the formation of a user community promoting the effective use of the library. We developed a system named *Shizuku2.0*, which shared the reading histories of the library other users who had the same interests as a focal user. This mechanism offers a cooperative reading environment.

Cooperative Reading

Reading books in the library is a form of learning that utilizes a document. The learning process basically traces the progress of a user in searching and gathering resources, acquiring knowledge, and utilizing that knowledge in real life. This learning process comprises a cycle in each user's information behavior.

The trace of the learning facilitated by such reading is recorded in the form of library use history, such as circulation data. Harada (2009) attempted to use the circulation data to implement a recommendation function for users, to aid their selection of books in a library system. A trial to improve library service by employing library usage data has already been performed, but the range of practical use is limited to the recommendation of documents or the evaluation of library service.

A mutual help network in the library delivers greater benefit if library users share not only *what books they selected* but also *how to understand the content of the book or how to utilize the book*. For example, the library tells a novice programmer how similar users made use of library resources, and enables such users to communicate with each other. As a result, these users will use the library services more effectively.

In this paper, we call library support that goes beyond the recommendation of resources as *support of cooperative reading*. *Cooperative reading* is a method of reading books that enables sharing of the reading process between readers who are interested in a common topic and their support of one another in that reading.

FEATURES OF SHIZUKU 2.0

Architecture

Shizuku2.0 is based on the service-oriented architecture (SOA). It is not a full-fledged library system in itself but works together with other library systems, such as the Integrated Library System (ILS), through a Web application programming interface (API). Therefore, libraries can install Shizuku2.0 without replacing their existing systems.

The detailed architecture of Shizuku2.0 is shown in Figure 1. It consists of three layers: the database layer, the component layer, and the user interface layer. Each layer has three functions: the reading stream function, the user profile generation function, and the reading stream unification function.

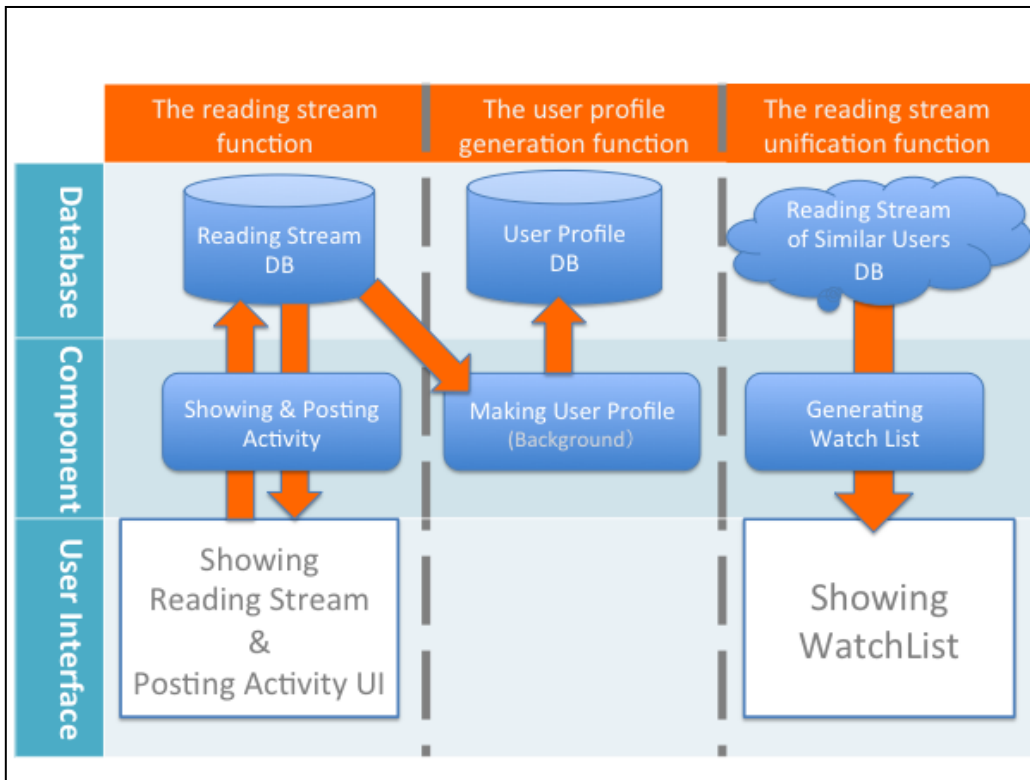


Figure 1: Architecture of Shizuku2.0



Figure 1 : User Interface of Shizuku2.0

User Interface

There are three parts to the home screen of Shizuku2.0 (Figure 2): the Reading Stream, User Bookshelf, and Watch List. The Reading Stream is the reading history of the user, the User Bookshelf is the cover list of books that the user is reading or has read, and the Watch List is the integrated Reading Stream of multiple users, recommended by the system. The user of Shizuku2.0 mainly updates the Reading Stream and browses the Watch List.

Reading Stream

From the users' reading materials, Shizuku2.0 semi automatically records their reading history and distributes it to other users. We refer to this history as the Reading Stream, and it consists of Activities, which are the smallest units that describe the users' reading activities. More specifically, Activities include (1) records of library use (i.e., records of books borrowed, search terms entered in the OPAC, etc.) or (2) notes taken while reading. With regard to (1), Shizuku2.0 provides a plug-in to automatically extract the above-mentioned records from the ILS. The unit of extracting an Activity is a search query or a book that the user circulates. On the other hand, with regard to (2), Shizuku2.0 provides a Web-based interface via which the users can easily take notes while reading. The length of a note is limited to 140 characters, similar to the micro blogging site, Twitter.

Recommending Users

In forming a reader community, the ease with which one can find readers interested in a similar topic is an important factor. However, it is hard to find such users efficiently by oneself, because (1) search costs increase in proportion to the size of the community; (2) psychological costs rise in proportion to the number of readers that the user does not know. Sometimes the users do not notice that their friend is participating in the community. Therefore, it is necessary for the system to recommend similar readers to users, in order to facilitate formation of the reader community. In this paper, we propose a technique for recommendation of similar readers using Nippon Decimal Classification (NDC). We describe the details of this technique in the section entitled NDC Tree Profiling.

WatchList

It is necessary to provide a communication tool in which users advise each other to push forward, reading smoothly while grasping other's reading progress.

Watch List is a function that allows one to gather other readers' reading streams on Shizuku2.0 by registering recommended users. In the Watch List, we can watch other members' activities and communicate by making comments on the activities of each other.

Whenever a user loads a home screen, the Watch List is revised through the subsequent handling of three phases.

1. Gathering the list of recommended users from the user profile database
2. Gathering the reading streams of the recommended users from a reading stream database
3. Displaying the unified reading stream to a home screen, sorted in chronological order

NDC TREE PROFILING

NDC Tree Profiling is a technique to analyze users' reading streams and to profile their tendencies in book readings, supporting implementation of the Recommending User function. Concretely, NDC Tree Profiling extracts NDC codes included in the reading stream of each user and constructs a hierarchical structure based on the data, constituting a user profile. Shizuku2.0 recommends similar users by calculating the similarity of these hierarchical structures.

NDC is a book classification rule established by the Japan Library Association. An NDC code consists of three columns of numbers and a decimal. Each figure shows the division of the hierarchical structure into three levels and aids in performing the division. For example, for a

code of 913.6, the first column value of 9 expresses *literature*; the second column value of 1 refers to *Japanese literature*; finally, the third column value of 3 refers to a *Japanese novel*.

Furthermore, an NDC code makes it possible to estimate the upper subject and the adjacent subject, because NDC is a classification system having a hierarchical structure. This characteristic implies that the structure of the NDC tree expresses the breadth and strength of the interest of the user. When few reading streams of similar users are available, the NDC codes can be used to achieve reliable recommendation in terms of relevance to the focal user. Therefore, the NDC code may resolve the "cold start" issue in collaborative filtering, i.e. the user recommender system with NDC codes need not have large user base.

For the above reason, in this paper, we propose a means to create user profiles based on extraction of the NDC code and consideration of the tree-structured data. In this paper, we refer this tree structure as an NDC Tree.

Process of Generating User Profile

First, it is necessary to build a user's NDC tree, to establish their user profile. The NDC code of each book is extracted from the user's reading stream. In particular, this action entails the following three steps. First, the program extracts and lists all the ISBN codes in the user's reading stream activities. Second, the program obtains bibliographic records corresponding to the ISBN codes from the bibliography database through an API that is offered by the National Diet Library digital archive portal (PORTA). Finally, the program extracts and totals NDC codes from the bibliographic records.

Second, the program creates the NDC tree from the list of NDC codes. This stage consists of three sub stages: node conversion of the NDC code, formation of the root node, and insertion of the node. With respect to node conversion of the NDC code, the program divides the NDC codes into digits, drops the decimals, and converts them into tree nodes. For instance, a NDC code of the form 913.6 is converted into three labels: 9**, 91*, and 913. All NDC codes was divided into three labels and was merged into one tree. For instance, when a user has three NDC codes: 410.02, 421, 913.02 and 914, NDC tree is shown as Figure 3.

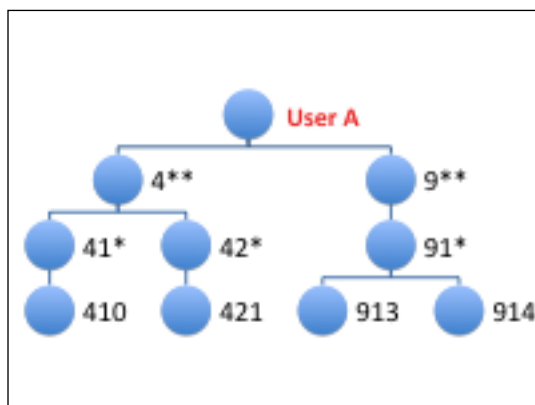


Figure 2: An example of NDC Tree

The Calculation of the Similar Degree between NDC Trees

As a method of calculating the degree of similarity between NDC trees, we use a combination of three parameters: tree edit distance, common label score, and comment rate.

The tree edit distance is the indicator of difference of a couple of trees. It defined as the sum of the cost of inserting or deleting nodes from a tree to achieve the same structure as another tree. When the tree edit distance is a small value, this implies that the structures of two trees resemble each other. In addition, it makes it possible to incorporate the difference in content between trees when considering operations to change labels. Therefore it is possible to calculate the degree of similarity between the widths of interest of users by measuring the editing distance between their NDC trees. In this paper, we use the Zhang&Shasha algorithm to

calculate the tree edit distance (Barnard, 1995). The score of tree edit distance between user A and B is calculated by following formula.

$$D_{T_A, T_B} = (d_{\max} - d(T_A, T_B)) / d_{\max} \dots\dots\dots(1)$$

T_Xuser X's NDC trees,

$d(T_A, T_B)$the tree edit distance between trees A and B, and

d_{\max} the maximum of tree edit distances.

The common label score is a means of calculating the similarity of the subjects represented by the NDC trees, based on a weighted score accounting for the importance of the nodes. Concretely, this approach searches for common labels (NDC numbers) between trees from the top-level on down for each hierarchy. The score is then based on the total weighted summation of the number of the searched labels. The common label score between user A and B is calculated by following formula.

$$C_{T_A, T_B} = \sum_{i=1}^3 w_i c_i(T_A, T_B) \dots\dots\dots(2)$$

$c_i(T_X, T_Y)$ the number of common nodes of layer i ,

between trees X and Y and

w_ithe score given to each common nodes at layer i .

We assume that the helpfulness of a user's reading stream to another user can be determined by the rate at which the another user makes comments, because these comments is helpful to understand why a book was read by another user. In this paper, the comment rate is utilized as one means of calculating user similarity, with the goal being that a recommendation result serves as better reference for a user. The score of comment rate of user B is calculated by following formula.

$$R_B = w_R (\text{comment}(B) / \text{total}(B)) + 1 \dots\dots\dots(3)$$

$\text{comment}(X)$ the number of comments in user X's reading stream,

$\text{total}(X)$ the number of activities in user X's reading stream, and

w_R the weight of the comment rate (0~1).

For the above means of calculating similarity, $\text{sim}(A, B)$, the similarity between user A and user B, is calculated by following formula.

$$\text{sim}(A, B) = D_{T_A, T_B} C_{T_A, T_B} R_B \dots\dots\dots(4)$$

This expression prevents noise in the recommendation. For instance, if the number of common nodes between user A and user B is large, but user B is reading a lot of books on a subject in which user A is not interested, the reading stream of user B likely becomes mere noise for user A. It is possible that the similarity between combinations of users that present significant noise for one another can be evaluated as low by taking the edit distance into consideration.

EVALUATION

Evaluation Plan

To generate a user recommendation using NDC Tree Profiling, we conducted an examination of 37 university student subjects using Shizuku2.0 for 18 days. We first requested that they register the books they had recently read. Next, we requested that they browse other examinees' reading streams and select five users that they felt would be helpful to their future reading. In this paper, we use this ranking data as the relevant at a set for producing recommendations. In the experiment stage, the user IDs in Shizuku2.0 are prepared at random to remove any influence of existing friendship as experimental noise.

In the above experiment, the user recommendation function was evaluated based on the reading streams dataset. First we obtained a ranked list of recommended users for each subject, based on the NDC Tree Profiling method. Next, we calculated the recall and precision

of the result. We also prepared the results of a random recommendation provision, as the baseline.

There are four parameters, namely w_1 , w_2 , w_3 , and w_R , used in calculating the similarity between users, as shown in equation (2) and equation (3). We tuned these parameters for use with our method to gain the maximum recall and precision. We do this by investigating a four-dimensional parameter space (w_1, w_2, w_3, w_R). Each combination of parameters is represented in this paper as follows: param (w_1, w_2, w_3, w_R). Recall and precision were calculated referring to (Brangier, 2007) and the final evaluation value is the mean value of the recall and precision of all users.

Result

Content of Reading Stream Dataset

There is a large difference in the number of registered books among the examinees. One examinee registered 272 books, while a second examinee registered none. In addition, the reading tendency of all examinees was strongly biased toward "literary" subjects.

Evaluation Result for Recommendation Algorithm

The result of the experiment is shown in Table 1. In this table, the rank of all parameter patterns is listed alongside its corresponding recall and precision values. The maximum recall was 0.3556 and the maximum precision was 0.2778, obtained with param(1, 2, 1, 0). There are 73 parameter patterns where the recall and precision were maximum values.

Table 1: Result of evaluating recommendation using NDC Tree Profiling

Parameters Pattern	Recall	Precision
param(1, 2, 1, 0)	0.3556	0.2778
param(1, 3, 0, 0)	0.3556	0.2778
param(1, 3, 1, 0)	0.3556	0.2778
param(1, 3, 2, 0)	0.3556	0.2778
param(1, 5, 0, 0)	0.3556	0.2778
param(1, 1, 1, 0)	0.3444	0.2685
param(1, 1, 2, 0)	0.3444	0.2685
param(1, 2, 0, 0)	0.3444	0.2685
Recommendation at Random	0.1715	0.1722

DISCUSSION

The Factor of Evaluation Results

The maximum recall and precision for recommendations produced using NDC tree profiling is 0.3556 and 0.2778, as shown in Table 1. Comparing this performance to a random recommendation, which resulted in a recall of 0.1715 and a precision of 0.1722, we consider the recommendation produced using NDC tree profiling to be effective. However, since the performance of a conventional information recommender system is higher, the performance of our method requires further improvement.

We found that the performance of our method improved when w_2 is larger than other parameters. This means that the second level of the NDC code could be the most effective

expression of users' interests. On the other hand, the comment rate did not contribute to the performance of the recommendation system because parameter w_R was always 0 in those configurations exhibiting highest performance. In addition, the tree editing distance is a useful piece of information in generating user recommendations, based upon a comparison to those recommendation results obtained without it.

We analyzed two users who had a recommendation precision above 0.5. As a result, it was found that these two users had common nodes in their NDC trees that are not associated with *literature* (9**). In Figure 4, there are two NDC trees, one each for users A and B. In this figure, the common nodes are those that have been colored. The NDC code 594, implying *embroidery*, was common to users A and B.

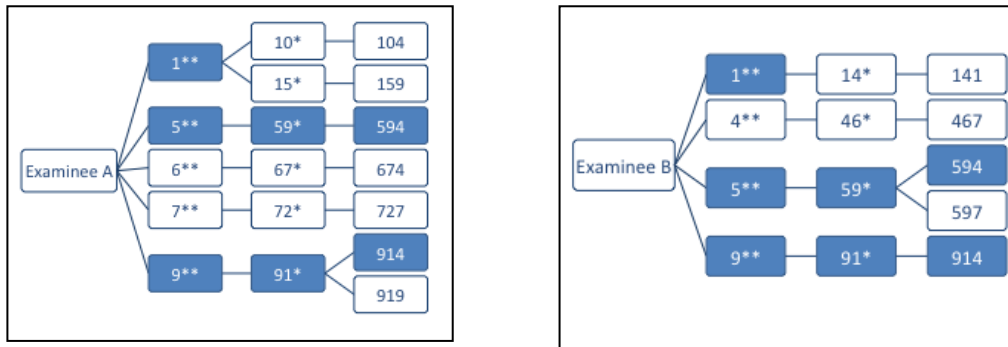


Figure 3: NDC Tree of Examinees A and B

On the other hand, we analysed recommendation results that were not correct and found that in addition to common nodes there was also a noise factor. In Figure 5, The NDC tree of user C has two nodes (594 and 914) in common with user A at the third layer. However, user C's tree involves many nodes associated with the subject of comics (726). Those users that user A identified as helpful did not register comics in their reading streams. The reason that user A did not select user C as a helpful user may be that user C registered many comics. To improve the performance of our method, we must incorporate a method by which users can control the weighting of the scoring method. For example, a user might be able to set a negative weight for those nodes that are not of interest to them.

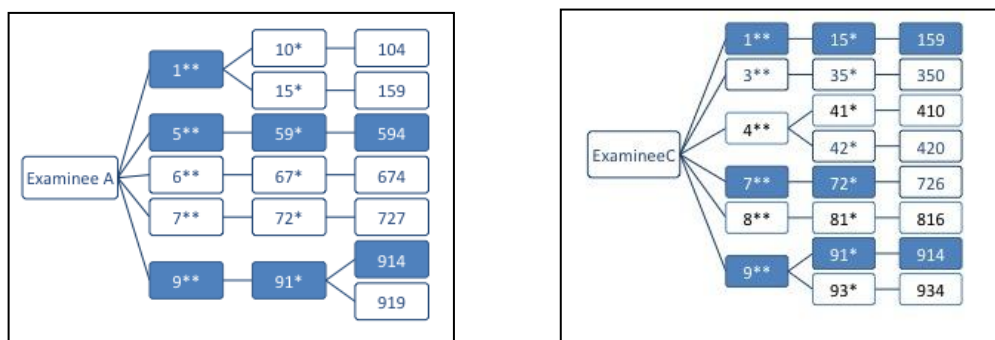


Figure 4: NDC Tree of Examinee A and C

User's reading tendencies in this experiment were focused mostly upon Japanese literature. We therefore consider the creation of a user profile based not only on the NDC but also on other characteristics of the user's reading stream. It is necessary to adopt a suitable technique that expresses every user interest. In addition, a hybrid recommendation model (e.g., our method and collaborative filtering) may improve performance.

Comparison to Existing Systems

Various Web services already exist where users can record the progress of their reading and communicate with each other. The most popular among these are Dokusho Meter (<http://book.akahoshitakuya.com/>), Media Marker (<http://mediamarker.net/>), and Booklog (<http://booklog.jp>). In Booklog, for instance, users can create their own bookshelves and make them public on the Web. In addition, they can use the bulletin board to communicate with each other and can register other users as friends, as in a social networking site (SNS). However, existing systems do not provide recommendations of other users having similar interests: if the user wants to make friends with people having similar interests, he or she must find them manually. Shizuku2.0 semi automatically identifies similar users and promotes cooperative reading for learning.

CONCLUSION

In the present study, we proposed cooperative reading, which is a reading support technique allowing library user to support reading for study by another library user. To achieve cooperative reading, we developed Shizuku2.0, in which users shared their reading history with each other. To recommend users who have the same interests, efficiently, we proposed and developed NDC Tree Profiling as a technique of generating user profiles on which to base suggestions. To evaluate our method, we performed an experiment. As a result, we found that a recommendation produced using NDC Tree Profiling is more effective than that obtained by random selection. In addition, it appeared that the second division of an NDC code could be most effective in expressing users' interests. However, there is room for improvement in the performance of this system.

NDC Tree Profiling can be applied to the book recommendation, i.e. recommending books in similar class. We will examine the method to apply NDC Tree Profiling to the book recommendation.

The prototype of Shizuku2.0 is now available on the Web. Moreover, we will develop APIs for Shizuku2.0 so that various types of libraries can introduce Shizuku2.0.

REFERENCES

- Barnard, D., Clarke, G., & Duncan, N. (1995). Tree-to-tree correction for document trees. *Technical Report*, 95-372. Retrieved from <http://ftp.qucis.queensu.ca/TechReports/Reports/1995-372.pdf>
- Blyberg, John. (2007). AADL.org Goes Social. blyberg.net. Retrieved from <http://www.blyberg.net/2007/01/21/aadlorg-goes-social/>
- Brangier, E., Dinet, J., & Eilrich, L. (2009). The 7 Basic Functions of a Digital Library - Analysis of Focus Groups about the Usefulness of a Thematic Digital Library on the History of European Integration. *Human Interface and the Management of Information. Designing Information Environments*, 5617, 345-354.
- Harada, T. (2009). The book recommendation system using library loan records. *Digital Libraries*, (36), 22-31.
- Herlocker, L., Konstan, J., Terveen, L., & Riedl, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5-53. Retrieved from <http://doi.acm.org/10.1145/963770.963772>
- Inoue, S., & Hori, Y. (2007). A trust model of social networking services and their applications for libraries. *IPSI SIG Technical Report, 2007-FI-88(109)*. Retrieved from <http://ci.nii.ac.jp/naid/110006531780>
- Murakami, K. (2007). Is it the next-generation library service? – What is Library 2.0?. *Current awareness-E*, 291. Retrieved from <http://current.ndl.go.jp/ca1624>
- Nagata, H. (2008). New 'Ba'(locale) in academic libraries: Information commons and learning commons. *The Annals of Nagoya University Library Science*, (7), 3-14. Retrieved from http://libst.nul.nagoya-u.ac.jp/pdf/annals_07_02.pdf