Building Semantic Web Applications as Information/Knowledge Sharing Systems

Hideaki Takeda[1,2] and Ikki Ohmukai[1]

[1] National Institute of Informatics (NII)
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan
[2] The Graduate University of Advanced Studies (Sokendai), Japan

Abstract. In this paper, we propose the methodology to design Semantic Web applications that can be acceptable widely by ordinary people. We first analyze "miracle of web" as an information sharing tool that is basically difficult for people to accept. In order to overcome this point, Semantic Web applications should have two types of gratification simultaneously, i.e., instant gratification that can be obtained even without information/knowledge sharing, and delayed gratification that can be obtained through information/knowledge sharing. The gap between two types of gratification can be bridged by the *translucence strategy* that lures people into information/knowledge sharing by showing delayed gratification within kissing distance. We then show our experience to build information/knowledge sharing tools with the above methodology. One is Community Navigator that helps participants for a conference to share knowledge like topics and related people. The other is Semblog systems that helps weblog people to exhibit and exchange their information more.

1 Introduction

Researchers tend to be obsessed with technical details when solving problems, i.e., they tend to forget the purpose or mission of the problem itself. It likely happens more when technical problems looks complicated and difficult. Semantic Web is probably the case. There is a nice technical road map like "the layer cake" and each step looks challenging technologically. According to the road map, many technologies have been developed like RDFS and OWL. But we are not sure how these technologies would contribute the purpose of the original problem. We start with this viewpoint.

Web is no doubt an information sharing tool. Scientists have been eager to exchange data and information among their organizations and communities quickly and easily Web have firstly spread out to people in universities. Then ordinary people find out that web is also useful for them, and expand web for their use. We have so accustomed to life with web, but wide dissemination of web is probably "miracle of web", because "**people are basically reluctant to exhibit their information.**"

Recall our daily life. We are even reluctant to put a free ad paper on wall in a supermarket nearby. Before web, there exist information publishing tools like ftp, but only limited people used such tools. Without intention to exhibit information to others, information sharing can not work. Information sharing is basically a difficult task to involve people.

As successor of web, Semantic Web should be an information sharing tool. Of course Semantic Web is going one step beyond web, i.e., aims to be a knowledge sharing tool. So it is reasonable that Semantic Web will be more difficult than web for dissemination.

We should develop Semantic Web applications carefully to involve people to use them as information/knowledge sharing.

The paper is organized as follows; In the following section, we show our methodology to build information/knowledge sharing systems. Then we show two systems we built in the following two sections (Section 3 and 4). In Section 5, we dicuss other information sharing systems and conclude the pager in Section 6.

2 Double-loop Gratification

We can enumerate many benefits for information/knowledge sharing, while there exist also hurdles for dissemination of information/knowledge sharing. One of the hurdles is the privacy and security issue that is related to sociological point of view.

Another is the feedback issue that is related to cognitive point of view. The feedback on contribution to information sharing is rarely visible. One of the reason why people do not wish to use information sharing tools is that their effort looks in vain because of lack of feedback. McDowell et al. [1] pointed out this issue as *instant gratification*. They said that instant gratification is needed to involve people in Semantic Web applications, and their application called Mangrove have succeeded because of realization of instant gratification.

We agree to importance of instant gratification, but instant gratification should be different in information/knowledge sharing applications. In Mangrove, users' contribution is quickly reflected to information sharing results by collecting and revising revise them as fast as it can. It is a nice feature but it sacrifices variety and scalability of information/knowledge sharing, because information/knowledge sharing takes time naturally.

We think that information/knowledge sharing applications should have two types of gratification simultaneously, i.e., instant gratification that can be obtained even without information/knowledge sharing, and delayed gratification that can be obtained through information/knowledge sharing. It always takes efforts for users to be accustomed with new applications. Instant gratification can be an anchor to keep users to use the applications. While users keeping to use them, delayed gratification that are real benefits of information/knowledge sharing arrives in them. The balance of two types of gratification is important rather than quantity of them. As I mentioned above, benefits from information/knowledge sharing tends to take time, it is too strict restriction to require instant gratification by information/knowledge sharing.

Web has both types of gratification. Authoring hypertexts gives people instant gratification. It is a new fascinating method for people to organize own information that is difficult to write down as stable well-organized form like word processing documents. Since authoring hypertexts and publishing them are so closely connected in Web, people are publishing their information with almost no extra efforts. Then they will receive delayed gratification as feedback from users who read their published information.

The problem is how to design such systems with two types of gratification. Through our observation on other systems and our experience on information/knowledge sharing, we propose *translucence strategy* to make people to shift instant gratification receivers to delayed gratification receivers. The strategy is simple: just put people in a situation where they can feel possible delayed gratification within kissing distance. Then they shift to the next step where they can receive delayed gratification. The step should be minimum, i.e., it should be a very small amount of extra efforts to join information/knowledge sharing in addition to ordinary efforts to obtain instant gratification.

In the following section, we explain two systems we built and how the above strategy works on them. The same methodology was applied to other systems like Ba-log[2] that stimulates communication based on location with location-embedded weblog and Social Scheduler[3] that assists people to determine schedules of shared tasks by analyzing personal network.

3 Community Navigator: Collaborative Scheduling Support System for Conferences

We built a system called *Community Navigator* that supports conference participants by helping their own scheduling *and* communication among them[4].

The first look of the system is just a personal scheduling system for a conference (see Figure 1). Users can browse the timetable of the conference and detail of each session and paper, and click bottoms to slot in papers they like to listen. Then the system shows their personal schedule both as a timetable and a list of papers.

But the system has another function, i.e., information sharing and recommendation by interpersonal network. When users browse paper pages, they can also click authors' names to register them as their acquaintance. The system also shows a list of "know" people and "is known by" people in the personal scheduling pages (see Figure 2). The former means a person whom the owner of the personal scheduling page actually registers as acquaintance, and the latter means person who registers the owner of the page as acquaintance. After people register acquaintances, they can access detail information of their acquaintances and receive recommendation of papers and people from the system that calculates the degree of importance among their acquaintances.

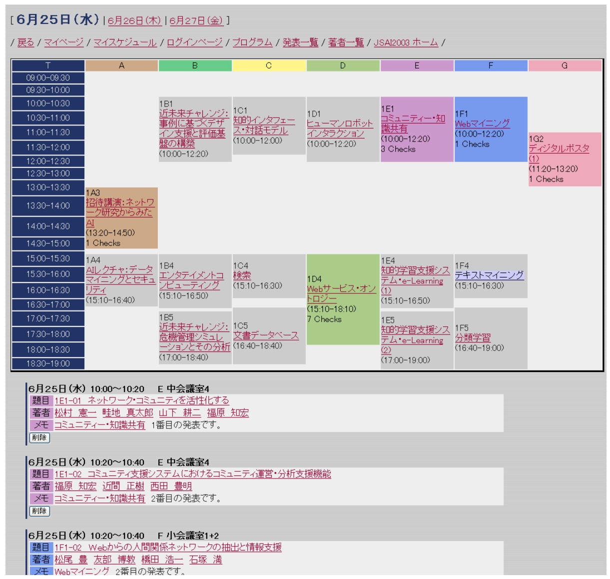

Fig. 1. Community Navigator: The first look is just a scheduler

We applied the system to an academic conference called *JSAI2003* and the result is very successful.

The conference held three days in 2003. 259 papers were presented and about 400 people were participated in the conference. The system was used by 276 users, and among them 160 users added 1840 papers in their schedules and 99 users registered 840 persons as acquaintance. These are significant numbers as acquirement of users, because there are no obligation to use the system. About 40% of participants actually used the system, and about 60% among them stepped forward to sharing information stage.

In this system, instant gratification corresponds to personal scheduling function, and delayed gratification to information sharing via interpersonal network. Personal scheduling function successfully attracted people to use the system. Our *translucence strategy* here is that we require minimum action like clicking their acquaintance and the system then starts information sharing with information already registered as personal scheduling. It is noted that the rate to enter the information sharing stage is 60%. We think the number is very successful but even with such a strategy, about a half out of initial users are involved in the information sharing stage. It suggests that involving people in the first stage as many as possible is important.

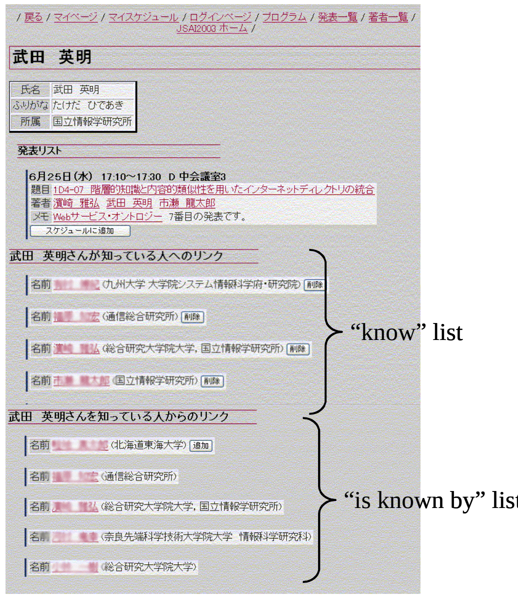

Fig. 2. Community Navigator: It can work as a navigator of community

4 Semblog: Metadata-driven Personal Publishing System

In this section, we introduce a personal knowledge publishing system called *Semblog* that provides an integrated environment for distributing small contents and making human relationship seamlessly [5]. It enables people to exchange information and knowledge with easy and casual fashion in degrees of personal interest, e.g. checking, clipping, and posting with various metadata and Weblog tools.

We developed two types of RSS aggregator called "RNA" and "glucose".

4.1 RNA: Web-based RSS Aggregator

RNA is a Web-based RSS aggregator written with Perl. A user can operate RNA through her/his Web server. Figure 3 shows a snapshot of RNA.

Firstly the user should register URIs of RSS in configuration page of RNA shown in Figure 4. The user can categorize these RSSs. List of sites checked by the user are converted into an RSS that can be used by other RSS-based applications again. RNA can also import and export OPML that is a standard metadata set for Web bookmark.

RNA produces site/entry list ordered by updated time of each element. After getting RSS files from various sources, RNA parses these RSSs and merges into single a "global" RSS tree. RNA converts this global tree to several forms by ordering chronologically. These partial trees are published as RSS and rendered into HTML.

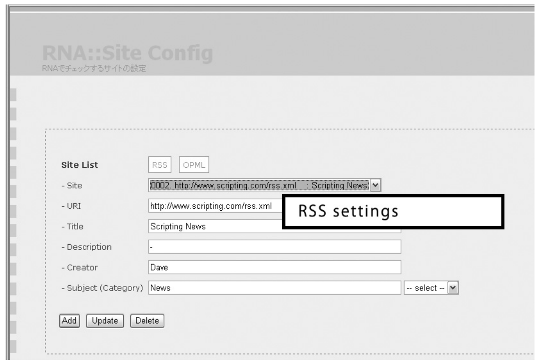

Fig. 4. RSS Registration

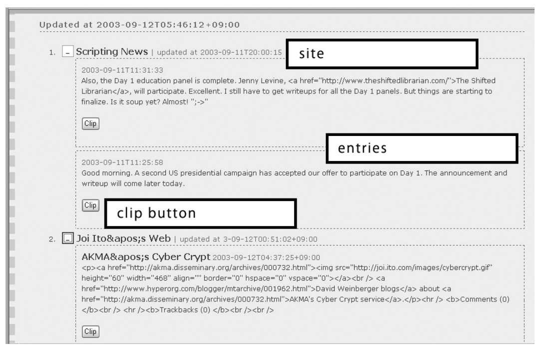

Fig. 5. Site List in HTML

RNA checks syntax of acquired RSS and corrects them if they are not valid. RNA converts all versions of RSS into 1.0, which is based on RDF model.

4.2 Glucose: Stand-alone RSS Aggregator

Glucose is also an extended RSS aggregator for Windows. Figure 6 shows a snapshot of Glucose. Unlike orthodox aggregators, Glucose is developed to support information distribution process in cooperation with coordinating with RNA. Main functions and interfaces of Glucose are shown below.

Like in RNA, the user registers URIs of RSS or OPML site list. Glucose can access several news sites without RSS by "sensor" script which extracts articles and converts them into RSS.

Glucose has three panes interface. The left pane shows "RSS Channels" which is subscribed by the user. The upper right pane indicates the headline list of contents including title, updated time, source and category. The lower right pane shows original contents.

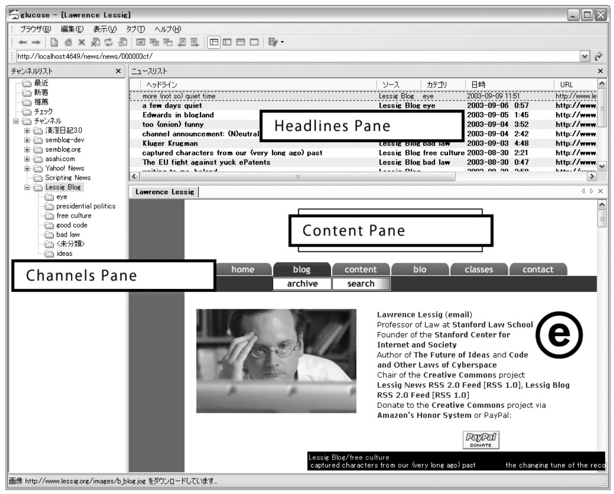

Fig. 6. Snapshot of Glucose

Glucose can extract TrackBack links from each content. Obtained links are shown below the corresponding entry in headline pane with "Re:..." like a mailer.

With Weblog editor interface (Figure 7) in Glucose, the user can post an entry to her/his Weblog if she/he has strong interest for content. This interface uses XML-RPC protocol. The user can clip contents to the clip list of own RNA using XML-RPC, then clipped contents are published via RNA.

4.3 Double-loop Gratification with Semblog

In our system, instant gratification is realized by the basic functions of RNA and Glucose. People simply read RSS-based contents from various information sources with our aggregators. These functions can make benefit to individual users in reading and writing Weblog contents.

Clipping is one of instance of translucence strategy in Semblog. For herself/himself, clipped content works as reminder that means "what I was thinking about?" instantly. On the other hand, someone who browses her/his clips can understand "what she/he was interested in?" because all clips are published on the Web.

Our Semblog system can be used as an information sharing platform. It is based on simple metadata so that it can be extended easily. We develop a new type of recommendation and retrieval systems to support delayed gratification as follows.

FOAF TrackBack Each RNA has XML-RPC interface that can send and receive its data dynamically RNA alliance is a content recommendation system based on cooperation of multiple RNAs.