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.

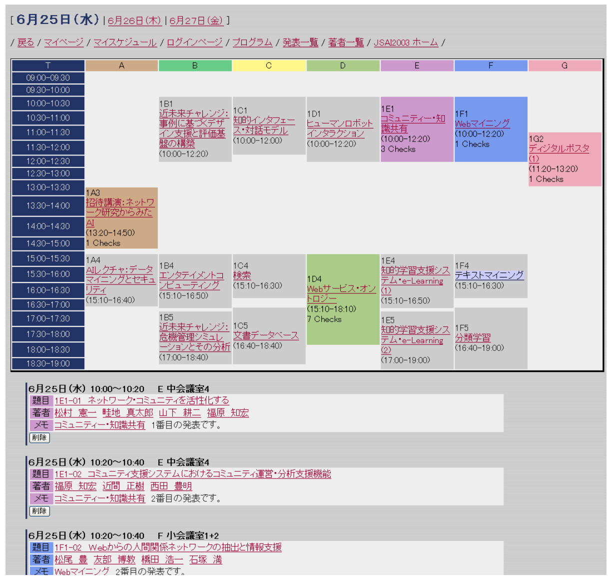

Fig. 1. Community Navigator: The first look is just a scheduler

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.

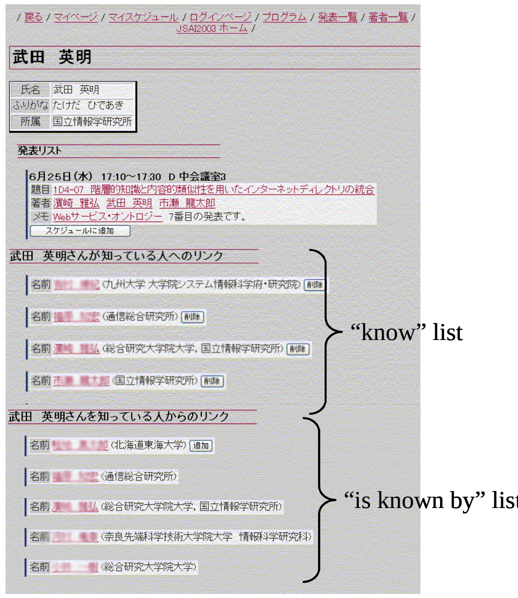

Fig. 2. Community Navigator: It can work as a navigator of community

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.

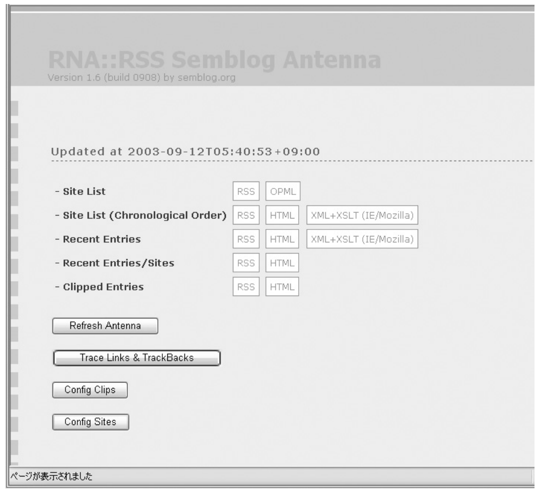

Fig. 3. Snapshot of RNA

Figure 5 shows a site list with HTML. User can browse description element of RSS in each channel (site).

RNA supports various output formats such like HTML, RSS and other forms i.e., JavaScript with server/client-side XSLT engines and original template engine. The user can create customized partial tree using plug-in and template script.

The user can save a favorite content in RNA to a clip list with one click. Clipped contents are stored in the "clipped" RSS tree and it is published like other RSS. RNA can post clips to social bookmark service such like del.isio.us[3].

RNA extracts TrackBack links from each entry in registered sites, and embeds TrackBack metadata in RSS and renders it.

The current version of RNA cooperates with RSS-based search engine such as Technorati[4] and Bulkfeeds[5]. By setting some keywords, the user can obtain new contents from non-registered sites.

Most of RSS generated by Weblog and news sites includes categorical information with `<dc:subject>` vocabulary. RNA can aggregates only specific contents by selecting certain categories and re-distribute categorical RSS.

It is necessary to get RSS and build trees occasionally since those contents are changeable with update of information sources. RNA can update periodically by cron interface of the server. Update interface can be called both manually and remotely by XML-RPC message that is generated automatically by Weblog tools.

[3] http://del.isio.us/
[4] http://www.technorati.com/
[5] http://bulkfeeds.net/

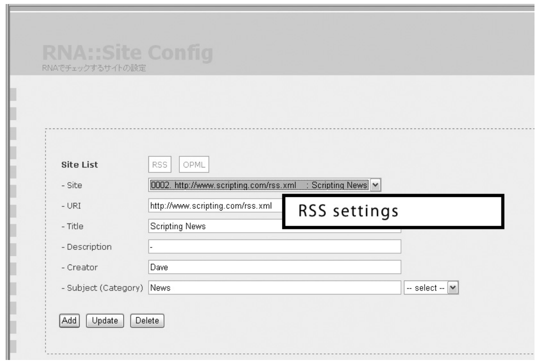

Fig. 4. RSS Registration

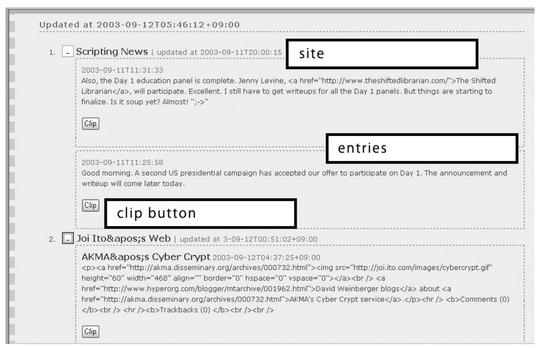

Fig. 5. Site List in HTML

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 lowerright pane shows original contents.

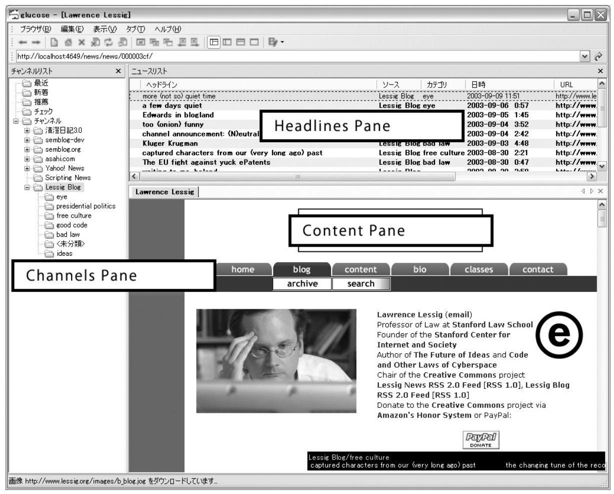

Fig. 6. Snapshot of Glucose

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.

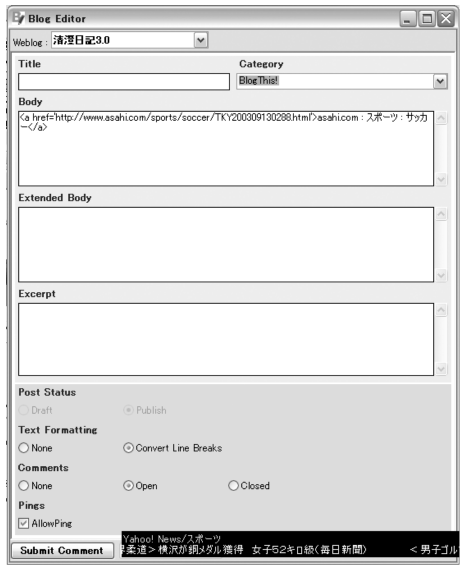

Fig. 7. Weblog Editor

We use FOAF metadata to identify each RNA. FOAF is RDF-based metadata format for describing human relationship. Besides the basic elements such as name, email and URI of the user, FOAF provides a statement that user X knows user Y.

The current version of RNA can generate FOAF data. RNA also has an interface for FOAF management to extend social network easily. We call this method as "FOAF TrackBack".

First the user X enters an RNA URI of the user Y in her/his own FOAF manager. The manager X asks the manager Y to acquire the FOAF data of Y, and writes "X knows Y" link in its FOAF. The manager Y records "Y isKnown by X" link in its FOAF and notifies to the user Y. If the user Y agrees, her/his manager registers "Y knows X" link. Repeating this process, a personal network of the user is constructed. Following recommendation methods are performed in the network.

RNA Alliance RNA Alliance is collaborative recommendation based on difference of registered sites or clips among multiple RNAs. At first it calculates similarities: S_i between the user's RNA:R_0 and a RNA on the personal network: R_i $(1 < i < n)$. Each RNA has a list of URIs: $R_i = \{u_0, \ldots, u_k\}$.

$$S_i = \frac{|R_0 \cap R_i|}{|R_0| + |R_i|}$$

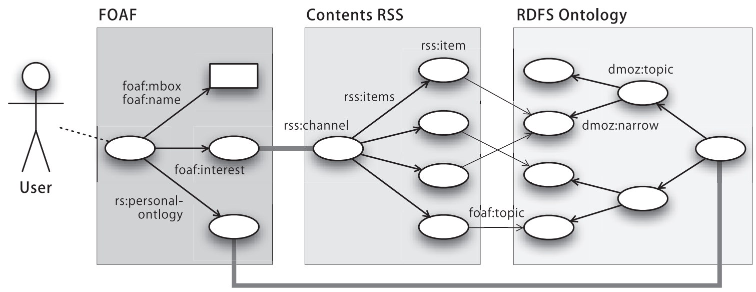

Fig. 8. Personal Ontology Framework

The system gives recommendation score: $V(u)$ to each URI by the following formula:

$$V_i(u) = \begin{cases} S_i & \text{if } u \in R_i \\ 0 & \text{if } u \notin R_i \ (i = 1, \ldots, n) \end{cases}$$

$$V(u) = \frac{\sum_{i=i}^{n} V_i(u)}{n}$$

This score is used for recommendation to R_0's user if URI u is not included in R_0.

The system shows the list of recommended URIs sorted by the score. The user can add these URI to her/his own "check" list.

Personal Ontology We propose a bottom-up personal ontology framework using RSS and FOAF metadata. To process small contents in various forms, we have to annotate a semantic markup with an ontology language to those contents. It is difficult to organize practical ontology hierarchy with top-down approach because building and maintaining such well-organized large ontology takes a lot of efforts. We aim to develop loose and bottom-up ontology system by combining personal classification, because we consider that personal knowledge will be represented with a routine work such as categorization and arrangement of information. Figure 8 indicates a conceptual architecture of the personal ontology system.

At first we define a personal ontology as a hierarchical system of categories. Everyone has those categories, and they routinely classify described and collected contents to the category. A label of a category can be named arbitrarily by user.

Unlike the conventional ontology, the personal ontology has to be related to the person who produces it. Therefore we apply FOAF metadata to link between the ontology and the person.

Personal ontology metadata consists of FOAF, RDFS Ontology and Contents RSS. The FOAF describes personal information, and the RDFS ontology shows

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:rs="http://www.roughsemantics.org/rs/0.1/"
>

<foaf:Person>
<foaf:name>Ikki Ohmukai</foaf:name>
<foaf:nick>i2k</foaf:nick>
<foaf:mbox rdf:resource="mailto:i2k@grad.nii.ac.jp" />
<foaf:weblog rdf:resource="http://www.semblog.org/i2k/" />
<rdfs:seeAlso rdf:resource="http://www-kasm.nii.ac.jp/~i2k/foaf.rdf" />
<foaf:interest rdf:resource="http://www-kasm.nii.ac.jp/~i2k/index.rdf" />
<rs:personalontology
          rdf:resource="http://www-kasm.nii.ac.jp/~i2k/ontology.rdf" />
....

</foaf:Person>
</rdf:RDF>
```

(a) Extended FOAF

```
<RDF xmlns:rdf="http://www.w3.org/TR/RDF/"
     xmlns:dc="http://purl.org/dc/elements/1.0/"
     xmlns="http://directory.mozilla.org/rdf">

<Topic rdf:id="Top">
  <tag catid="1"/>
  <dc:Title>Top</dc:Title>
  <narrow rdf:resource="Top/Arts"/>
  <narrow rdf:resource="Top/Business"/>
  <narrow rdf:resource="Top/Economy"/>
  <narrow rdf:resource="Top/Tech"/>
  ....
</Topic>

<Topic rdf:id="Top/Arts">
  <tag catid="2"/>
  <dc:Title>Top/Arts</dc:Title>
  <narrow rdf:resource="Top/Arts/Fine"/>
  ....
</Topic>
```

(b) RDFS Ontology

```
<item rdf:about="http://www.semblog.org/i2k/archives/000304.html">
<title>Blog Hacks</title>
<link>http://www.semblog.org/i2k/archives/000304.html</link>
<description>
Monday's child is fair of face, Tuesday's child is full of grace,
Wednesday's child is full of woe, Thursday's child has far to go,
Friday's child is loving and giving, Saturday's child works hard for his living,
And the child that is born on the Sabbath day is bonny and blithe, and good and gay. ...
</description>
<dc:subject>trivia</dc:subject>
<foaf:topic rdf:resource="http://www-kasm.nii.ac.jp/~i2k/ontology.rdf#Top/Arts">
<dc:creator>i2k</dc:creator>
<dc:date>2004-04-09T01:24:16+09:00</dc:date>
</item>
```

(c) Contents RSS

Fig. 9. Personal Ontology Metadata

a structure of the categories, and the contents RSS shows written and collected contents by the user.

We add two elements to basic FOAF model shown in Figure 9 (a). One is `<foaf:interest>` which is to point the contents RSS, and the other is `<rs:personalontology>` that is originally defined by our Rough Semantics project[6] to indicate the RDFS ontology.

The RDFS ontology is described with the form of Open Directory RDFS format shown in Figure 9 (b). Each node has a fragment ID.

The contents RSS is similar to a conventional RSS. Our RSS uses `<foaf:topic>` to point a category on the RDFS ontology, while the conventional model applies `<dc:subject>` to express a thesis of a content. This makes our RSS to have backward compatibility. Example of this RSS is shown in Figure 9 (c). It should be noted that topics pointed by this tag are not restricted to those in their own ontology, rather any topics in others' and some global ontology. Separating ontology and instances enables such flexible management.

FOAF, RDFS ontology and RSS are described in separate files so that we can keep compatibility with existing applications on these formats. This is a

[6] http://www.roughsemantics.org/

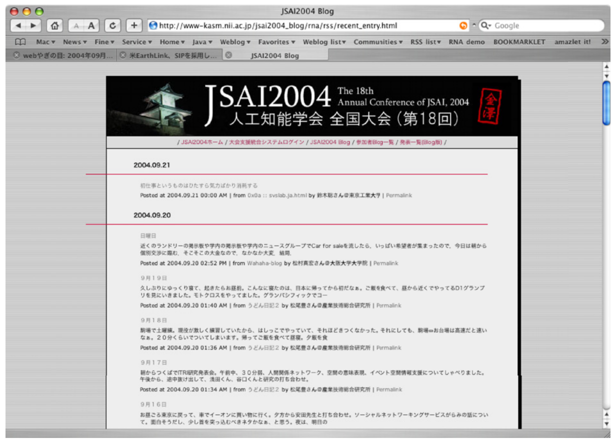

Fig. 10. RNA in an academic conference

great benefit that our system can cope with such existing applications via these files.

Our framework enables applications and services to produce new types of search or recommendation. For example, mapping methods between two directories or bookmarks are applicable to the personal ontology. Egocentric search[6] is also able to realize easily by building a social network with `<foaf:knows>` in the users' FOAF.

Unlike these peer-to-peer model, we can calculate a similarity among a personal ontology and the global ontologies such like WordNet and ODP in advance. Multiple personal ontology can be matched with each other via the global ontology and this method needs less computation cost. In addition, it is not necessary to modify that algorithm in P2P model and personal-global model because both ontology has the same structure.

4.4 Use Case

We applied our system to some communities.

One is for an academic conference called JSAI2004 (Figure 10). Participants registered URI of her/his Weblog to RNA so that other attendees can browse various opinion for the conference and papers. Unlike conventional closed system, RNA provides that an author of an opinion keeps her/his authorship permanently.

Other example is education support. Senshu University developed class support system based on RNA. In this system, all students and teaching staff should have Weblog and all contents will be aggregated with each class or project respectively. The user post her/his content using original editor interface which communicates multiple Weblog tools and RNA. RNA aggregates and shows recently updated contents of member so that the user can access newest topics in the class and project in the university.

RNA is used as person-based contents management system. Research institute of economy, trade and industry (RIETI)[7] publishes Weblog of its research associates with RNA. Official contents should be managed in single policy but it may restrict their contribution since it is so messy to follow. On the other hand, it does not seem to official contents when each member just publishes her/his Weblog freely.

Thus RIETI introduces RNA to aggregate all contents from their Weblog and embed composite contents into official Web site. This model may decreases management cost.

We distribute RNA and Glucose in our web site[8]. About 3,000 users downloaded RNA and over 150,000 users downloaded Glucose from September 2003.

5 Discussion

We have seen a lot of failure of information sharing. But some exceptionally seem successful.

In one sense, one of the most successful information sharing is Amazon.com[9]. In Amazon.com, users' actions searching products are stored without any extra operations and used to recommend products similar to current actions. It is passive but powerful information sharing since users' actions are shared and used.

Another new trend for information sharing is *social tagging* like del.icio.us[10] and flickr[11]. We can say that it is another kind of bookmark sharing that could not be so popular. The difference is that it is more convenient in making and posting bookmarks and more powerful in using collected information.

They indicate that improving instant gratification is effective to involve people. On the other hand, improving delayed gratification is not easy to realize. One of the reason is that delayed gratification varies in communities. For example, density of information sharing also varies. Some community welcomes distributed style of information sharing like weblog with TrackBack and comments, and some embraces centered style information sharing like wiki. We should develop information sharing systems that fit types of information sharing they need.

We designed Semblog systems to allow users to select levels of information sharing in order to deal with variety of information sharing. We provide various levels of information sharing, i.e., posting clips, FOAF relationship, and personal-network based recommendation. We expect that users can find their appropriate level of information sharing by using Semblog systems.

7 http://www.rieti.go.jp/en/index.html
8 http://www.semblog.org/wiki/?en
9 http://www.amazon.com
10 http://del.icio.us/
11 http://www.flickr.com/

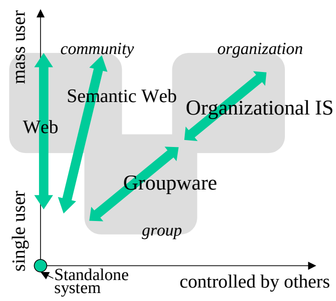

Fig. 11. Systems for community, group and organization

6 Related Work

Groupw arehas faced the similar problem from its beginning since acceptance by people is crucial. Grudin[7] summarized the eight challenges for Groupw are, including “disparity in work and benefit”, “critical mass and Prisoner’s dilemma problems”, and “disruption of social processes”. Some of them are also challenges for Semantic Web.

The difference betw een Groupw are and Semantic Web is ballance betw een control and mass of users (see Figure 11). Grudin placed Groupw are bet w een standalone systems and organizational information systems like management information systems. Standalone systems are single-user and no control by other people, while organizational information systems are multi-user and controlled by authority or other people. Groupw areis middle-scale in mass of users and partially con trolled. Web is, on the other hand, on the different line. Web is large-scale in mass of users but no control by authority or other people. The difference is what kind of people is target for systems. Web is used in community in which people are loosely connected and just share interest, while Groupw are and organizational information systems aim to support group and organization in which people are tightly connected and share common goal.

Semantic Web is slightly con trolled because maintenance of metadata is needed, but still by far free from control in comparison with Groupw are. Since suc h difference and similarity exists betw een Groupw are and Semantic Web, some solutions for Groupw are are applicable but others not.

We can pick up some lessons from Groupware research. One is supporting of social aw arenessthat tells people what other people do vaguely. The early example of realization of social aw areness is Babble system that shows interaction among people by location of points and circles. Another example is “Gleams of P eople”[8] that illustrates how people are active or communicate each other by blinking of balls. This approach can be used to realize translucence strategy since it may make people aw are ofimportance of participation to community gradually.

7 Summary

In this paper, we discuss how we can build Semantic Web applications appealing to ordinary people. We showed that two types of gratification should be needed, 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 also showed that the above methodology seems to work in building our applications.

I focus on information sharing rather than knowledge sharing because even information sharing is still a difficult task. More discussion is needed especially for delayed gratification that people can receive only after they are involved in rich knowledge sharing systems. I think that there is no royal road, but at least ballance between formality and familiarity in knowledge representation is crucial to realize widely acceptable knowledge sharing systems.

References

1. McDowell, L., Etzioni, O., Gribble, S.D., Halevy, A.Y., Levy, H.M., Pentney, W., Verma, D., Vlasseva, S.: Mangrove: Enticing ordinary people onto the semantic web via instant gratification. In: International Semantic Web Conference. (2003) 754–770
2. Uematsu, H., Numa, K., Tokunaga, T., Ohmukai, I., Takeda, H.: Balog: Location-based information aggregation system. In: Poster Proceedings of Third International Semantic Web Conference (ISWC2004). (2004)
3. Ohmukai, I., Takeda, H.: Collaborative task scheduling method based on social network analysis for cellphone application. In: Proceedings of the IADIS International Conference of WWW/Internet (ICWI2004), Madrid, Spain (2004)
4. Hamsaki, M., Takeda, H., Ohmukai, I., Ichise, R.: Scheduling support system for academic conferences based on interpersonal networks. In: Poster, HyperText 2004. (2004)
5. Ohmukai, I., Takeda, H., Hamasaki, M., Numa, K., Adachi, S.: Metadata-driven personal knowledge publishing. In McIlraith, S.A., Plexousakis, D., van Harmelen, F., eds.: The Semantic Web - ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Volume 3298 of Lecture Notes in Computer Science (LNCS). (2004) 591–604
6. Numa, K., Ohmukai, I., Hamasaki, M., Takeda, H.: Egocentric search based on RSS. In: Poster Proceedings of Third International Semantic Web Conference (ISWC2004). (2004)
7. Grudin, J.: Groupware and social dynamics: eight challenges for developers. Communications of ACM **37** (1994) 92–105
8. Ohguro, T., Yoshida, S., Kuwabara, K.: Gleams of people: Monitoring the presence of people with multi-agent architecture. In: Approaches to Intelligent Agents (PRIMA'99 proceedings). Volume 1733 of Lecture Nots for Artificial Intelligence. Springer-Verlag (1999) 170–182