

Blog を基盤とした情報流通システムの提案と実装

Proposal and Implementation of Information Sharing System based on Blog

徳永徹郎

山口仁

伊東匡

Tetsuro TOKUNAGA Hitoshi YAMAGUCHI Tadashi ITOH

日本電信電話株式会社 NTT ネットワークサービスシステム研究所

NTT Network Service Systems Laboratories, NTT Corporation

インターネットの普及により、様々な形態での個人からの情報発信が行われているが、それらのうちもっとも普及している Blog においてはエントリの記述は自由文で行われる。したがって、定型化することが可能な内容が定型化されずに記述されている例が多く見られる。これらを定型化することで多様な情報流通サービスが考えられる。これらの定型化可能な種類のエントリについて、RSS の拡張を行い情報の流通を活発にすることを目的としたシステムを提案、および作成した。本システムは、定型化可能な項目を定義した項目定義表を用いて、入力部分、データ生成部分、閲覧部分の各機能をカスタマイズすることを特徴としている。このシステムを実装し、オフィスでやり取りされる情報の中で扱う情報を対象に、小規模な試用実験を行った。そこから、いくつかの機能要望を得て、今後の課題の抽出を行った。

1. はじめに

近年、Blog の普及はめざましく、Blog を用いて発信するユーザが増加し、そのユーザたちに発信される情報量もまた増加している。一方、閲覧者は多量の情報から自分の求めている情報を探し出すことが困難になってきている。

例えば、ある映画の感想が書かれたエントリ (Blog における記事) が多くあったとしても、それらを一覧することはできない。また、例えば実際に鑑賞した感想なのか、それとも雑誌を見て行きたくなったという話が書かれているかなどエントリの内容種類については検索エンジンを利用しても、すぐにはしぼりこめない。また、今公開されている映画で、もっとも評判が良い映画はどれかといったことは知ることができない。

しかし、例えばクチコミ情報の交換サイトなどでは、

対象となる話題ごとに入力フォームが定まっており、それに従って情報が投入される。このことにより、各項目に対する検索等の操作が容易に実現でき、評判の良い映画を抽出するなど高度な検索を行うことが可能である。これは、特定の種類の情報を特定のフォーマットに従って入力することによって得られる効果である。

ところで、Blog で記述される内容には、様々なものがあるが、その中でも、書かれている内容を定型化することが可能な種類のエントリがある。例にあげた映画の感想などのレビューのエントリは、感想を述べる対象のものを紹介する際に、タイトルと監督名などが書かれることがある。これらの情報を特定のフォーマットに従って入力することができると、発信者の入力補助となりうる。また、これらの情報をエントリのメタデータとして扱うことが可能となれば、エントリの検索にも応用が可能となる。しかし、既存の Blog では、項目名と値を追加することが想定されていないため、現状の Blog が備えている機能だけでは、定型項目

連絡先: 徳永 徹郎、日本電信電話株式会社 NTT ネットワークサービスシステム研究所
〒180-8585 東京都武蔵野市緑町 3-9-11
tokunaga.tetsuro@lab.ntt.co.jp

を扱うには適していない。

これまでに筆者らは、このような定型化可能なエントリを対象として、エントリの流通を支援する方式のコンセプトを提案していた[1]が、本稿ではさらに検討を進め具体化した方式を提案し、プロトタイプ実装および、その試用結果を報告する。

2. 定型化する情報の特徴

定型項目を含んだ情報流通のシーンとして、映画の感想を書く状況と、閲覧する状況を例にして、エントリに含まれる情報の特徴を述べる。

まず、感想を書くにあたって、言及対象を特定する必要があるために、鑑賞した映画のタイトルを書く。これは、閲覧者にとっては、エントリ内容の対象を特定するために必要な情報である。また、映画の感想などのクチコミ情報を得るために、映画のタイトルをキーワードとして用いて検索することも多い。このようにエントリの言及対象を特定するキーは、閲覧者の検索のキーとしても用いられる。

閲覧者が検索の際にあると便利な情報には、これとは異なる種類の情報もある。例えば、映画の感想を検索する際には、総合評価としてお勧めできるのかどうかを数値でエントリに付加されていると、好評価のエントリ、低評価のエントリを並べ替えて表示するなどのより便利な閲覧が可能となる。

また、漠然とした検索を行いたいシーンもある。例えば、おもしろいコメディ映画を観たい場合、映画作品のジャンルとお勧め度の情報がエントリに含まれていれば、Blog から検索することが可能となる。映画

作品のジャンルは言及対象のメタデータであり、お勧め度はエントリのメタデータとして必要である。

これらのことから、定型化すべき項目として、言及対象を識別する情報および付加情報、およびエントリの内容を表すための付加情報があげられる。

3. 要求条件

本章では、前章であげた定型項目情報を含んだエントリを発信し、閲覧するシステムに求められる要求条件を挙げていく。

(1)同じ種類の情報は、エントリごとに共通の項目を持つ

まず1点目として、同じ種類の情報は、共通の項目を持つ必要がある。エントリごとに項目が異なるとは、他エントリとの比較などができずに本方式のメリットが享受できない。多くのユーザが共通の項目を利用して、エントリを記述することで、項目ごとに検索する効果が高くなる。

(2) 分散した複数記事から項目ごとに検索を行うなどの操作が可能

次に、共通の項目を持つエントリに対して検索を行うことが可能となるようにする必要がある。特定のBlogだけではなく、分散した複数のBlogのエントリを対象として検索可能とする必要がある。

(3) 異なる種類の定型項目を持つエントリをひとつのBlog上で利用可能

異なる項目を持つエントリについても、ひとつの

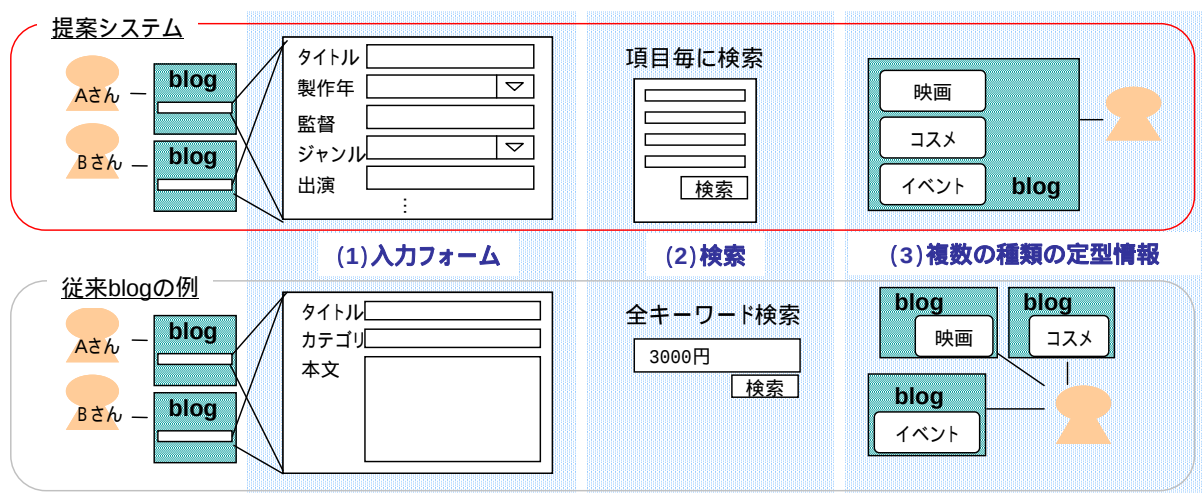


図 1 要求条件

Blogに記述することができるように、複数の種類のエン
トリーに対応する必要がある。また、システム提供者
から考えると、ひとつのシステムで複数の種類のエン
トリーを生成、閲覧することで、速やかなサービス提供
ができる。

既存のBlogツールでは項目名とその値を追加す
ることが想定されておらず、これらの要求条件を満た
す機能は提供されていない。

提案システムでは、項目を考慮して流通情報のデ
ータ構造を新たに設計することで、正確な検索処理
を実現する。また、各項目を入力画面に表示するこ
とで、何を書いてよいかわからない初級発信者に対
しては、入力すべき情報を提示することで、情報発
信の支援を行う。また、閲覧者の求めている情報に
近づけるためにより詳細な情報発信を促すといった
効果が期待できる。

4. 類似サービス・技術

・ コミュニティサイト

Blog 以外に、定型項目のある情報を発信するサー
ビスとしては、コミュニティサイトで行われているレビ
ューの投稿などがそれにあたる。例えば、化粧品な
どのコスメ用品のクチコミ情報を交換するサイト[2]で
は、定まったフォームに従って情報が入力され生成
されるため、項目ごとに検索をかけることなどが容易
である。しかし、コスメのクチコミ情報を専門に扱うた
め、他のクチコミを行うことはできない。

Blog 関連では、エントリーの整理を行うためのサービ
スがいくつか実現されている。

・ トラックバックの利用

Blog の標準的な仕様であるトラックバックを利用し
てエントリーの言及対象に基づいた整理を行うサービ
スがある。トラックバックを用いて、特定のテーマにつ
いてのエントリーを集めるサービスである。例えば、書
籍毎にトラックバックを受け付ける URL を用意するこ
とで、書評エントリーを集めることが行われている[3]。こ
れを用いると、言及対象(本)ごとに書評エントリーを
閲覧することができる。しかし、お勧め度などのエン
トリーの内容を表すための付加情報を定型化し入力する

ことはされていないため、それに基づいたエントリー
の検索等は行えない。

・ Tagging

RSS 検索エンジンのテクノラティ[4]では、エン
トリーのメタデータとなるキーワードを、エン
トリーの著者が特定の記法で記述しておく
と、テクノラティのクローラがそのメタデータを
解析して、RSS 検索エンジンで活用することが
実現されている。これは、発信者がキー
ワードを定義するため、発信者の主観が反映され
たメタデータが付加される。また、入力すべき
項目の提示がされないため、閲覧者の必要とす
るメタデータが付加されない可能性がある。

・ hReview[5]

様々なエントリーにメタデータをつける試みの
ひとつである Microformats の中のひとつ
であり、各種レビュー記事に XHTML を用
いてメタデータを付加する方式として提案され
ている。hReview は、メタデータの記述方式
のみの提案であり、その方式のメタデータ生
成方法については、特に規定されていない。

・ Structured Blogging[6]

Structured Blogging はエントリーの構
造化を行う方式として提案されている。HTML
の中に構造化されたデータの XML を記述す
る方式である。これもメタデータ生成方法に
ついては、特に規定されていない。

5. 提案方式

5.1. システムの特徴

本システムでは、従来の Blog と RSS リー
ダによる情報発信、閲覧方式に対して、以下
の機能を加えることにより実現する。

- (1)各種情報の定型項目を定義した項目定義表を利用する
- (2)RSS を拡張したものを交換データとして利用する
- (3)項目定義表を参照して、入力ツール、閲覧ツールのユーザインターフェースなどを変更する

これらの特徴を有する本システムの概要は図 1 に示す。これらの特徴について以下に詳しく述べる。

5.2. 項目定義表の外部参照

定型化可能な項目を定義した項目定義表を作成し、システム内に設置する。ただし、項目定義表は、入力機能、ファイル生成機能、閲覧機能、検索機能等の各機能に影響のある部分なので、各機能とは独立させて設置する。また、複数のユーザが参照できる

入力機能やファイル生成機能は、Blogツールをベースに、外部に存在する項目定義表を動的に参照し、適切な入力画面や、適切なファイルを生成する機能を追加する。これによって、要求条件(1):「同じ種類の情報は、エントリごとに共通の項目を持つ」を実現する。

閲覧用のRSSリーダの表示部や検索フォームについても、項目定義表を参照して変更する機能を持たせる。

また、項目表の外部参照という形をとることで、複数の種類の項目表をひとつのBlogツール、RSSリーダで扱うことが可能となり、要求条件(3):「異なる種類の定型項目を持つエントリをひとつのBlog上で利用可能」を実現する。

項目定義表には、各項目の名前や値の種類、値が選択式であれば選択肢など、入力時、検索時の各項目のフォームを定義する値が定義される。項目定義表間での連携やオントロジーなどは現時点では考慮していない。

5.3. RSS の拡張

エントリ毎に、定型項目についての情報を追加するために、RSS などのサイトサマリにタグを追加する。これらは、もともと XML で記述され情報が構造化されており、拡張は容易である。本提案方式では、このファイルを流通させるデータととらえて扱う。

拡張された RSS の一例を図 2 に示す。追加する情報は、本方式による拡張を示す<catac>、後述する今回実装したシステムで提供した「周知」の項目定

```
<item rdf:about="http://blog.nslab.ntt.co.jp/tokunaga/archives/2005/05/news4.html">
<title>今日のニュース</title>
<link>http://blog.nslab.ntt.co.jp/tokunaga/archives/2005/05/news4.html</link>
<description>
<![CDATA[ 種類:周知<br> 重要度:普通<br> <!-- comment --><a href="http://www.ntt-
xx.co.jp/xxx/xx_xxx.html">IT業界ニュース</a><br>
たいへんおもしろいサービスが開始されています。
]]>
</description>
<dc:subject />
<dc:creator>tokunaga</dc:creator>
<dc:date>2005-05-17T12:26:21+09:00</dc:date>
<catac:genre>notification_request</catac:genre>
<catac:notification_request>
<notification_request:type>notification</notification_request:type>
<notification_request:priority>0</notification_request:priority>
<notification_request:comment>
<a href="http://www.ntt-xx.co.jp/xxx/xx_xxx.html">IT業界ニュース</a>
たいへんおもしろいサービスが開始されています。
</notification_request:comment>
</catac:notification_request>
</item>
```

図 2 RSS の拡張例

義表を利用していることを示す<notification_request>で構成される。

また、拡張したRSSに適應するRSSリーダを閲覧ツールとして利用する。RSSリーダでは、これまでに取得したRSSのキャッシュを利用して複数のBlogに対する検索などの処理を行うことができる。このようにして、要求条件(2):「分散した複数記事から項目ごとに検索を行う」などの操作が可能となる。

6. 実装システム

本章では、提案方式を実現するために実装したシステムの概要を説明する。

6.1. 機能概要

機能概要図を図 3 にあげる。情報が発信され閲覧されるまでの流れを追って、各機能を説明する。

Writer 機能を用いて、情報発信者は入力を行う。入力項目で必要となれば、商品情報 DB(データベース)を利用して入力の補助を行う。生成されたデータは Blog システム機能に送られ、DB に保存され、HTML や拡張 RSS が生成される。Viewer 機能は、登録されている拡張 RSS を取得してキャッシュ DB に蓄積する。閲覧者はキャッシュされたデータから検索などの処理を行いながら閲覧する。

システムの動作環境としては、各機能をひとつのサーバー上で構築した。OS は Windows XP SP2 で、Writer、Viewer は CGI として動作し、Perl で記述されている。Blog 用 DB、キャッシュ DB、商品情

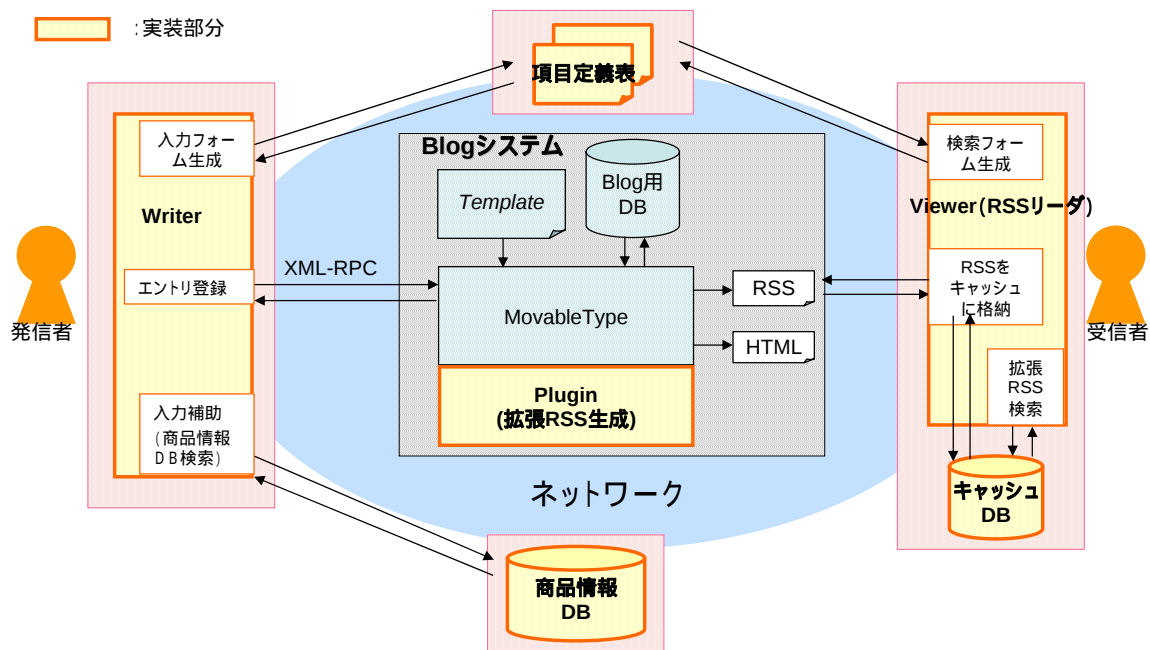


図 3 機能概要図

報 DB は、MySQL を利用した。

変更した。

6.2. 項目定義表

項目定義表で定義する項目は、その情報の種類によって異なるが、今回は各項目を以下の3種類に分けて、取り扱った。

- (1)言及対象物について(商品など)
- (2)発信者の行動について(状況など)
- (3)発信者の思考について(感想・評価など)

これらの種類によって、入力時の画面表示の順序を制御した。

また、画面生成のために、HTML のフォームに対応するよう、Text、Select、Radio などの値を各項目の属性として定義した。そして、Select や Radio の場合は、選択肢も項目定義表内部にも記述した。

6.3. Blog システム

Blog システム機能の Blog エンジン、MovableType[7]を利用した。MovableType はプラグインによる拡張が可能であるため、プラグインを用いて、RSS を生成する際にテキストフィルタをかけて、RSS の拡張に必要な文字列置換を行った。また、HTML を生成する際には、ブラウザから閲覧しても見やすい形になるように、定型項目部分を普通のエントリの本文とは異なると認識できるようにデザインを

6.4. Writer

Writer で特徴的な機能は、項目定義表を参照して入力フォームを生成する機能である。実際の画面のスクリーンショットを用いて説明を進める。

最初に、これから書こうとするエントリの種類を選択する。これによって、用いる項目定義表が選択される。そして最初に、言及対象の情報を入力する。ここで、言及対象の情報が DB に格納されている場合は、DB 検索フォームを表示する(図 4)。検索フォ

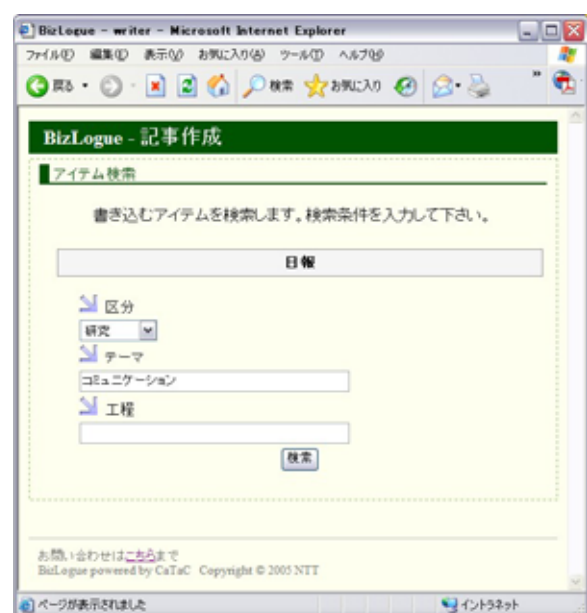


図 4 検索画面

ームは、項目定義表に書かれている言及対象物についての項目と同一である。次の画面ではそのフォームに入力された文字列から検索した結果を表示し、ユーザはそこから選択することで、言及対象の情報入力を完了する。

続いて、発信者の行動についての入力フォームを表示し、最後に発信者の思ったこと考えたことを入力するフォームを表示する(図 5)。

6.5. Viewer

Viewer では、CGI として動作する RSS リードとして実装した。使うためには取得対象の RSS の URL を事前にキャッシュ DB に登録しておく。閲覧者のボタン操作をトリガーとして、Blog エンジンが出力した拡張 RSS を取得し、キャッシュ DB に格納する。キャッシュ DB の中では、項目定義表毎にテーブルを生成し、蓄積する。取得した拡張 RSS が新しい項目定義表であれば、項目定義表を参照し、テーブルの定義を行う形で実装した。

図 5 本文などの入力画面

表示する時には、Blog 毎に表示する方法(図 6)と、項目定義表毎に表示する方法を標準の画面として用意した。検索を行う場合は、閲覧者は検索する対象の種類を選択して、項目定義表を参照して生成された検索フォームから検索を行う。従って、異なる項目定義表をまたいだ検索は実装していない。

7. 実験

7.1. 実験内容

作成したシステムの利用実験を研究所内で行った。今回、実験用に提供した項目定義表は「日報」と「周知」の 2 種類である。

日報は、ある時間からある時間までの業務を記録しておくエントリーと想定して項目を定義した。エントリーの言及対象の項目として、その時間に行った業務内容を選択し入力する。例えば、研修等の共通業務なのか、研究開発業務なのかを選択し、調査、検討、論文作成などの業務内容を選択する。また、エントリーに記述した業務を行っていた開始時間、終了時間も定型項目として用意した。

これらを利用することによって、業務中に得た情報や考えたこと、進捗状況などを共有化する効果が期待できる。また、どの業務をどれほどの時間行ったかを集計することによって、業務の振り返りや、各種事務処理に利用することもできる。今回は、ユーザごとに月ごとに集計し、表として画面に出力する機能を提供した。

2 つめに、周知エントリーの項目定義表を提供した。周知は、著者の属する組織においては、普段メーリ

図 6 任意の RSS を表示

ングリストで行われており、様々な情報が流れてくる。それらの周知情報には、情報共有が目的のものや、作業を依頼するものなどの種類がある。これらに共通する項目として、「締め切り」と「重要度」を定義した。締め切りには月日を入力し、重要度は星の数で0から3の間で選択し入力する。

これらから、今日が締め切りの作業を検索して表示したり、周知情報が発信された順番ではなく、締め切りの順番で並べ替えることなどが可能となる。

以上の2つの項目を定義し、研究所内の13人を対象に、1か月にわたって、システムを利用してもらった。この期間内に投稿されたエントリ数は123件であり、その内訳は、日報64件、周知59件となった。

7.2. 評価

ユーザから寄せられた、本システムについてのコメントから、簡易評価を行う。

まず機能要望として、「日報」については、「日報を見せる対象を指定できるようにしてほしい」、「周知」については、「周知対象を制限した周知が出来るようにしてほしい」という要望を得られた。

これは、今回実験した環境と、提供した項目定義表の特性であると考えられる。日報エントリは、自分の作業メモという位置づけで行うことが多いと推測されるためか、「異なる仕事をしているユーザに対して新着エントリとして表示されるのは気が引ける」という意見などがあつた。また、同ユーザから「Blogで公開されること自体はそれほど気にならない」という意見もあつた。このことから、Blogとしてイントラネット上に公開することと、新着エントリとしてRSSリーダに表示することは、公開意識に違いがあることが推測できる。

周知エントリについては、内容によって対象者が異なることがあっても、全ユーザに新着エントリとして表示されるため、同様に気が引けてしまうという意見などがあつた。

本システムの効果として、コミュニケーションの活性化が現象として見られた。これまでメーリングリストで行ってきたニュース情報の共有などを本システム上で行うことによって、議論が活性化した。メーリングリ

ストに返信することより、Blogへ「コメント」を付けることの方が気軽に行われていることが推測される。

また、利用環境の問題が判明した。常に立ち上げているメールのアプリケーションでは、周知は届くものであったが、本システムではブラウザでWeb上のRSSリーダサイトをチェックしに行かなければならない。その手間が煩雑であり閲覧者にとっては、不便である。また、情報発信者にとってもメールに比べ、「読まれている」という安心感がないこともわかった。

8. まとめと今後の課題

本稿では、Blogにおいて共通の定型項目を持つエントリにメタデータを付加することによって、情報流通を実現するシステムを提案した。

今後は、項目定義表のより詳細な記述方法の策定を進める。今回はテキストファイルで実装を行ったが、XML化して仕様として公開できる段階まで詳細を決めて行く必要がある。

また、入力の補助として位置づけている商品DBは、今回は独自に実装したDBであるが、今後はWebサービスなどでAPIが公開されているオープンなDBを利用することができるよう検討を進めていきたい。

そして、ユーザから得られた要望を基に、システムの使いやすさの向上をはかっていく。

また、メタデータの付加方法として、本システムで実装したRSSの拡張方式以外にも、類似研究であげた方式などこれから何が普及していくかはわからないが、これらの他の方法についても、システムの柔軟性を活用して、項目定義表をもとにしたメタデータ付加を行うことを検討していく予定である。

参考文献

- [1] 徳永 徹郎, 田中 洋平, 大石 哲矢, 黒川 章 : “Weblog を基盤とした制限情報を用いたコミュニケーションシステムの提案”, 電子情報通信学会通信ソサイエティ大会, B-20-3, 2004
- [2] アットコスメ <http://www.cosme.net/>
- [3] bk1 <http://www.bk1.co.jp/>
- [4] RSS1.0 <http://purl.org/rss/1.0/spec>
- [5] hReview

<http://developers.technorati.com/wiki/hReview>

[6] Structured Blogging

<http://www.structuredblogging.org/>

[7] MovableType <http://www.movabletype.jp/>