

Web アノテーション共有システム Cmew/U の設計と実装

廣津 登志夫[†] 高田 敏 弘[†] 青柳 滋 己[†]
佐藤 孝 治[†] 菅原 俊 治[†]

当初テキストや静止画の交換が主体であった WWW も、最近では音や動画像、三次元データ等、様々なメディアデータの交換の道具として利用されている。しかし、WWW による情報交換ではサーバに蓄積したデータの読み出しが中心で、本に対する書き込みのような情報の付加はあまり行われていない。本稿では、様々な種類のメディアデータに対して、アノテーション (Annotation: 注釈) の形での情報の付加を可能にするシステム Cmew/U の設計と実装について述べる。本システムはメディアごとに異なるアノテーション埋め込みの機能をモジュールとして実現することで、高い拡張性を提供している。また、性能評価の結果、インターネットの転送遅延に対して十分に小さいオーバーヘッドでアノテーション埋め込みの機能が実現できることを示す。

A Design and Implementation of Cmew/U Web Annotation Sharing System

TOSHIO HIROTSU,[†] TOSHIHIRO TAKADA,[†] SHIGEMI AOYAGI,[†]
KOJI SATO[†] and TOSHIHARU SUGAWARA[†]

In this paper, we describe the design and implementation of a multimedia Web annotation system that enables users to add information to multimedia contents obtained through the World Wide Web. We have developed a highly modular annotation server that stores users' annotations and embeds them into multiple types of contents including plain text, HTML documents, and MPEG video. The measured overhead on embedding annotations is enough small against the latency over the Internet. This system also provides a mechanism for sharing the annotations among multiple users using the same server.

1. はじめに

近年広域ネットワークを介した情報の交換が現実のものとなってきた。特に、World Wide Web (WWW) の登場により、大量の多様な情報に簡単にアクセスできるようになってきている。この WWW が情報共有に対してもたらした効果としては、

- ネットワーク上の資源を Uniform Resource Locator (URL) を用いて指定することによる容易な情報獲得、
 - マルチメディアコンテンツの容易な取扱い、
 - ハイパーリンクによる複雑な情報間の関連の実現、
- といった点があげられる。この複数メディアを統合し情報間の横断的な関連を付けるという機能に着目すると、WWW による電子的な情報メディアは本や雑誌といった物理的な情報メディアにはない優れた特性を

持つといえる。

しかし現在の WWW の環境では、ハイパーリンクの始点を配置できるのは HTML ファイルが静止画上のクリックブルマップに限定されており、また獲得した情報に対するリンクやメモといった情報の付加はあまり考えられていない。本や雑誌のような紙のメディアの場合にはユーザによるメモ書きが可能で、後から再度情報を閲覧する際に役に立ったり、他人の書き込んだ情報からの間接的な情報獲得が期待したりできる。そこで、この WWW においてリンクやメモといった情報を複数のメディアデータに対して付加する機能を実現し、さらにその付加された情報を複数ユーザ間で共有可能とすることで、WWW に紙のメディアの利点を加えることができると考えられる。これにより、たとえば同じ仕事をするプロジェクトのメンバー間で WWW を介して付加的な情報を共有することが可能になる。

著者らはこれまでに、音や動画像といった連続ストリームメディアを含む様々なマルチメディアコンテン

[†] NTT 未来ねっと研究所
NTT Network Innovation Laboratories

ツを統一的に取り扱うことができる Web ブラウジングシステム Cmew (Continuous MEdia with the Web) の構築を進めてきた^{1)~8)}。このシステムでは、複数のメディアデータに対してその中に始点を持つハイパーリンクを実現している。たとえば、動画画像のある時点での映像の特定部分に端を発するリンクや、音データの特定の区間に端を発するリンク等がその例であり、それらのハイパーリンク付きマルチメディアデータをブラウジングする環境も提供している。これは端的にいうと、従来の HTML テキストにおけるアンカー () タグや静止画に対するクリッカブルマップによるハイパーリンクを動画画像や音へ拡張したものである。

この Cmew システムにより、テキストや静止画のみならず音や動画画像を含む様々なマルチメディアコンテンツに対して、ハイパーリンクを渡り歩く形での統一的なアクセスが可能となった。しかし、このシステムは従来の WWW と同様な読み出し中心のシステムであった。今回、この Cmew システムを基本として、マルチメディアコンテンツに対してリンクやメモをアノテーション (Annotation: 注釈) として付加し、そのアノテーションを複数ユーザ間で共有することが可能なシステム Cmew/U (User Annotation) を実現した^{9)~11)}。このシステムでは、各々のメディアごとに異なるアノテーション埋め込み機能をモジュールとして実現し、複数のメディアに対する高い拡張性を実現するとともに、それらのモジュールを容易に開発するためのツールキットも提供している。本稿ではこの Cmew/U の設計および実装について述べ、アノテーション埋め込みのオーバヘッドを中心とした性能の評価結果を示す。

2. 背景

まず、本研究のベースとなっている Cmew システムの概要を述べる。さらに WWW に対する書き込みを実現した従来の研究事例について述べる。

2.1 Cmew Project

Cmew Project^{1)~8)} は、Web と統合されたマルチメディアコンテンツのブラウジング環境の構築を目指して 1997 年より著者らのグループで進められてきた。このプロジェクトでは『Web とマルチメディアコンテンツのシームレスな融合』を目指し、従来ハイパーリンクの終点にしかならなかったビデオ等のマルチメディアコンテンツを『HTML』や『クリッカブルマップの付いた静止画』等と対等に扱うために、連続メディアデータ中にハイパーリンクの始点を実現するためのフ

レームワークを提供している。Cmew Project で提供している主な機能としては、

- MPEG PRIVATE2 パケットを利用した、ビデオ中からのハイパーリンク (Cmew/M)^{1)~4)}
- 音の視覚化も含めた、音データからのハイパーリンク (Cmew/A)^{5),6)}
- Java Applet による各メディアデータの制御に基づくシナリオ作成機能 (Cmew/J)^{7),8)}

といったものがあげられる。Cmew/M と Cmew/A においては、位置・フレーム数・時間といった連続メディア中の時空間情報とリンクの終点の URL の対を、静止画のクリッカブルマップに類似した形式で記述し、それらを連続メディアの特定の部分に埋め込むことでリンク構造を実現している。これらは、単一のデータに固定的に埋め込まれたリンク情報を実現するものである。これに対して、Cmew/J のシナリオ機能ではメディアデータ自体にはリンク情報を保持する必要がなく、外部の制御ファイル (シナリオ) でリンク情報や複数のデータにわたる制御を記述する。これらのブラウジング環境は、SGI O2 上で稼働しており、部分的な機能は Sun Solaris 等にも移植されている。

2.2 WWW に対する書き込み

WWW で獲得したコンテンツに対してユーザが情報を付加するアノテーションの機能は、初期の Web ブラウザから提供されていた。NCSA Mosaic ブラウザでは、Web ページに対してアノテーションとしてテキスト情報をメモのように添付し、後から同じページを参照した際にその添付した情報を参照することが可能であった¹²⁾。また、このアノテーションをサーバに保持し複数のクライアントで共有する Group Annotations¹³⁾ の仕組みも作られた。しかし、この NCSA Mosaic の提供するアノテーションは Web ページ全体に対するもので、その内部の特定部分に対してのメモ書きやリンク付けといった注釈付けはできなかった。また、Group Annotations には、ユーザやグループに対するセキュリティの機能がなかったという問題があった。

Web ページ全体ではなく、中身の指定部分に対して情報を付加し共有するシステムとしては CoNote¹⁴⁾ や ComMenter^{15),16)} があげられる。これらのシステムは、テキスト中に印を付けて注釈を挿入しその情報をリンクのアンカーとして見せている。ここでは、アノテーションの共有やアノテーションの検索等の枠組みも提供されているが、対象が HTML テキストのみ

本来のメディアでは任意のユーザデータを入れてよい部分や末尾等。

で他のメディアには対応していない。また、これらのシステムに共通の問題としては、専用ブラウザを必要とし現在主流となっている Web ブラウザからはその機能が利用できないことがあげられる。

Web ページの内容の変更を支援することを旨とした研究として、WEBDAV (World Wide Web Distributed Authoring and Versioning) Working Group の活動があげられる^{17),18)}。これは、Web ページの内容をサーバ以外のクライアント上で変更することを目的としており、サーバ上での Web ページのデータのバージョン化や適切なバージョンを見せる枠組みを作ることを目指している。これは、現在標準化の進められている技術であるが、連続ストリームメディア等への対応までは考えられていない。

3. 設 計

マルチメディアコンテンツに対する Web アノテーションの機能を実現する Cmew/U システムの設計について述べる。本システムでは、連続ストリームメディアを含むマルチメディアコンテンツに対してアノテーションを付加し、さらに、それらのアノテーションを共有する機構を実現する。ここでは、同じ目的を持ったプロジェクトのメンバ等の限られたユーザ間での情報共有を支援することを想定して、比較的狭い範囲のネットワークにつながれた、数十～数百程度のクライアント間で情報付加を共有することを前提としてシステムを設計する。

まず、著者らの考える WWW に対するアノテーションについて整理する。次に、アノテーションの保持・提供を行うサーバを従来の Web アーキテクチャにどのように組み込むのがよいのかについて検討し、マルチメディアコンテンツに対するアノテーション付加機構の設計について述べる。最後に、サーバとクライアントの間で、アノテーションに関するユーザ等の属性情報 (アノテーション属性) を伝える方法について検討する。

3.1 Web アノテーション

一種の構造化文書である HTML テキストは、アンカー () タグによりリンク情報を記述することで文書間の関係を表している。また、静止画に対するクリッカブルマップは、静止画中の空間的領域と他の文書等の関係を表している。先に述べた Cmew システムでは、クリッカブルマップに類似した情報により、MPEG 等の動画像の時間的・空間的領域に対するリンクや、音データの時間領域に対するリンクを実現し、マルチメディアコンテンツ間の関連を

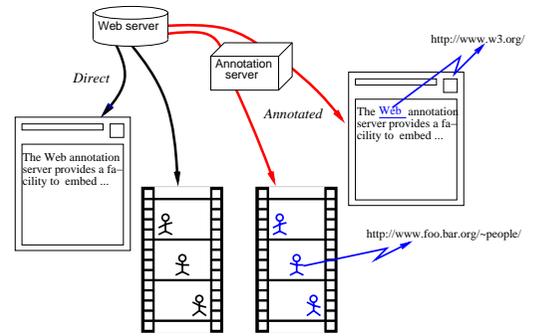


図1 Web アノテーション
Fig. 1 Web annotation.

表している。

本研究で著者等の目指している WWW への情報付加とは、直感的には本に対する書き込みと同様なもので、メモを付加したり、他の情報との関連を付けたり、余計な情報どうしの関連を消したりすることである。そのため、アノテーションの実体としては、これらのリンク情報自体もしくは、リンク情報とリンク先の情報 (たとえば、説明文のテキスト等) の組となる。したがって本研究における WWW に対する情報付加とは、Web サーバから獲得して来たコンテンツにリンク情報 (アノテーション) を埋め込むことである。図 1 に Web アノテーションの概念を示す。同一のコンテンツに対して、Web アノテーションシステムを介してアクセスした場合には元のコンテンツにないリンク情報が付加されている。

3.2 システムの構成

次に、アノテーションをどこに配置するのがよいかという点について検討する。WWW のアーキテクチャは単純化すると「コンテンツを提供するサーバ」! 要求と返答を中継したり、キャッシュしたりするプロキシ」! 受けとったコンテンツを表示するクライアント」の三部からなると考えられる (図 2)。以下に、各々の部分にアノテーションを保持した場合の得失について整理する。

(1) サーバ側

アノテーションが付加されるコンテンツを保持するサーバがアノテーションを保持すると、すべての要求に対してアノテーションを共有することが可能となる。しかし、ユーザごとのアノテーションを実現しようとする、全世界のクライアントを使うユーザを一意に認識する必要が出てくるので現実的ではない。また、アノテーションの機能を提供していない Web サーバには情報付加ができないので、情報に対する

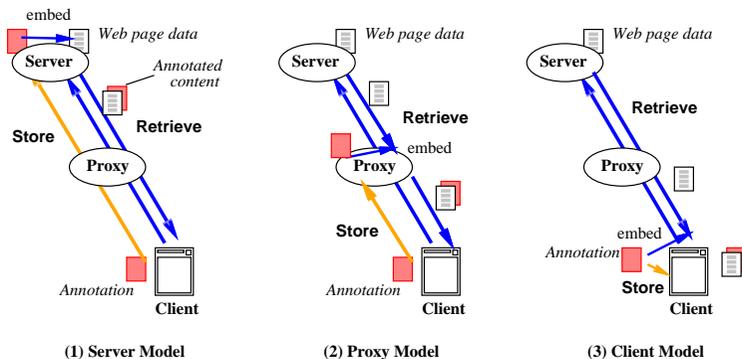


図2 アノテーションシステムのモデル
Fig.2 Three models of the annotation system.

アノテーション付けの範囲が制限されることになる。

(2) プロキシ

プロキシサーバがアノテーションを保持すると、そのプロキシサーバを利用するユーザの間でアノテーションを共有することが可能になる。ある程度の数のユーザに対してプロキシサーバを設定し、そこでユーザ管理を行えばよいので、(1)に比べて現実的である。また、複雑なアノテーションの埋め込み処理を行う場合には、次の(3)に比べてクライアント側の負荷を減らすことができる。一方、アノテーションを直接共有できるのは同一のプロキシを利用するユーザだけに限られ、異なるプロキシ間でのアノテーション共有には、別途プロキシ間の通信が必要となる。

(3) クライアント側

クライアントが付加したアノテーションを保持しておくと、ダウンロードしてきたコンテンツに対して、クライアント上でアノテーションの埋め込み処理を行うことになる。この場合、アノテーションを共有するためには、アノテーションの交換の枠組みを作る必要があり、クライアント上では「アノテーションを付加する元コンテンツ」と「付加するアノテーション」という複数のデータを同期的に読み込む機構を実現しなくてはならない。また、コンテンツの表示にある程度コストがかかる場合には、アノテーション埋め込みの処理が負荷になる。

このことから、アノテーション共有のためにはプロキシがアノテーションを保持する形態が得策であると考えられる。実際にアノテーションの共有を行う場合には、多くの場合はあらかじめユーザのグループをつ

くり、そのメンバの間で共有すると考えられるので、プロキシでメンバ管理をすればよいことになる。クライアントがアノテーションを保持すると、情報共有のためには結局何らかのサーバを実現する必要があるうえに、元コンテンツとアノテーションの両方のデータを並行かつ同期的に読み出す機構をクライアント上に実現する必要が生じ、システムの構造が複雑になる。そこで、本研究では Web プロキシサーバにおいてアノテーションを蓄積する形態で実現する。

3.3 マルチメディアコンテンツに対するアノテーション

アノテーションサーバは Web プロキシサーバの形で機能するもので、その基本的な働きは次の 2 点である。

- クライアントの要求する URL からコンテンツを獲得し、その URL に対応するアノテーションを付加してクライアントに返す。
- クライアントからアノテーションを受け取り、対象となる URL とともに保持する。

ここで、先に述べたようにマルチメディアコンテンツに対するアノテーションにおいては、その埋め込み方式や表現形式が多様であり、また新規データフォーマットやバージョンの違いクライアントの機能(ブラウザスペック)の違いに対処しなければならない。そのためには、アノテーション機能の拡張性が高いアノテーションサーバを構築する必要がある。そこで、「アノテーションの付加」「アノテーションの解析および蓄積」というアノテーションサーバの 2 つの基本機能をモジュール化し、動的な読み込みを可能にする(図 3)。ここでは、プロキシサーバに対して種々のアノテーションの機能を持ったモジュールをプラグインとして組み込むことにより、様々なデータタイプやブラウザスペックに応じた処理が可能になる(ブラウザに組み込まれ

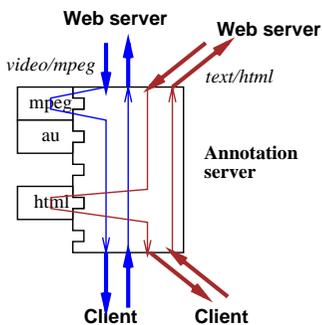


図3 アノテーションプラグイン
Fig. 3 Annotation plug-ins.

るプラグインと区別するために、サーバ側プラグインまたは Server-side Plug-in と呼ぶ)。これにより、アノテーション機能だけを単独で開発することが可能となるので、新しいデータタイプやブラウザスペックに対するアノテーション機能の開発も容易になる。

この、プラグインアノテーションサーバが Web サーバへのリクエストを中継する際の基本的な挙動は次のとおりである。

- (1) クライアントの要求から URL とアノテーション属性（ユーザ、グループ、ブラウザ情報等）を取り出す。
- (2) URL で指し示されているサーバもしくは他のプロキシサーバに対して要求を中継し、URL に対する情報を Web サーバから獲得する（プロキシとして機能する）。
- (3) Web サーバから得たコンテンツの MIME タイプや、要求から抽出したブラウザ情報を元に、適切なプラグインを特定する。
- (4) ユーザやグループの情報と URL から、埋め込むアノテーション（リンク情報）を検索する。
- (5) Web サーバから得たコンテンツとアノテーションをプラグインで処理した後、クライアントにアノテーション付きコンテンツを渡す。

3.4 アノテーションプロトコル

アノテーション付きのコンテンツをサーバから獲得するには、「誰がアノテーション付きのコンテンツを要求しているのか」という情報を要求のメッセージに載せなければならない。このために、クライアントとアノテーションサーバの間でなんらかのプロトコル拡張が必要である。アノテーションのために必要となる属性情報（以下、アノテーション属性と呼ぶ）としては、URL 以外に次の情報が必要になる。

- アノテーションモード
- ユーザ名と認証情報（パスワード等）

```
GET url HTTP/1.0 \r\n
Annotation: user&password="userinfo",
          group="ginfo",
          filtering keyword="keywords" \r\n
...Message headers ... \r\n
\r\n
```

図4 アノテーションヘッダ
Fig. 4 Annotation header.

- グループ名
- キーワード

アノテーションモードはアノテーションサーバへの要求が、アノテーション付きのコンテンツを要求している（“on”）か、アノテーションの付かない元々のコンテンツを要求している（“off”）かを表す。ユーザ名と認証情報はクライアントを利用しているユーザを認識し、サーバ上で各個人のアノテーションを探すために必要である。グループ名はアノテーションサーバ上でアノテーションを共有するグループを表す。キーワードはアノテーションサーバ上で適切なアノテーションを絞り込む検索のキーワードとして使用する。使用しているブラウザの種類やブラウザのプラグイン、Helper アプリケーションの種類の情報を持つブラウザスペックも、このキーワードの中に含まれる。

ここで、これらの属性情報をクライアントからアノテーションサーバに伝える手段として、次の3つの手法を用意した。

- アノテーションヘッダ
- URL 埋め込み
- 分離プロキシ

アノテーションヘッダは、HTTP（Hyper-Text Transfer Protocol）^{(9)~(21)}のバージョン 1.0 以降がサポートしている複数行のリクエストヘッダの一部として、アノテーション属性を送る。ここでは、拡張リクエストヘッダとして Annotation フィールドを作成し、そこにアノテーション属性を入れた要求を出す（図4）。この Annotation ヘッダは、クライアントとアノテーションサーバ（プロキシ）との間のみで使われ、Web サーバには伝わらない。アノテーションサーバは、この Annotation ヘッダのついた GET 要求をアノテーション付きコンテンツの読み出し、Annotation ヘッダのついた PUT 要求をアノテーション属性の保存として取り扱う。

URL 埋め込みでは、要求 URL の表現中にアノテーション属性を埋める。図5に例を示す。この方法では、要求 URL の一部がアノテーション属性を表しており、アノテーションサーバが URL を解釈することで、アノテーション属性が伝えられる。

```

http://annotation.server/Annotation/userinfo/
keywords/http/some.server.host/index.html
↓
mode="on", user&password="userinfo",
group="userinfo", filtering keyword="keywords",
URL="http://some.server.host/index.html"

```

図 5 URL 埋め込み
Fig. 5 URL embedding.

分離プロキシでは、2つのプロキシサーバを使う。1つは元々の Web サーバのコンテンツをそのまま伝える通常のプロキシで、もう1つはアノテーション付きのコンテンツを提供するアノテーションサーバである。このアノテーションサーバは、すべてのリクエストをアノテーション付きのコンテンツを要求しているリクエストだと見なして処理をする。各クライアントは、アノテーション付きのコンテンツを要求するかどうかに応じて、プロキシを使い分ける。この方式では、ユーザ情報やパスワードは HTTP Proxy-authentication を用いて伝える。

アノテーションヘッダの方法では、アノテーションヘッダ付きのリクエストを發するようにすべてのクライアントを変更する必要がある。しかし、元々のコンテンツを指す URL も、アノテーション付きのコンテンツを指す URL も同じなので、クライアントの挙動1つで「元々の(アノテーションの付かない) Web 空間」と「アノテーション付きの Web 空間」の間を自由に渡り歩くことができる。ユーザは「アノテーションのモードの変更」と「リンクのクリック」という2つの処理のこだけを考えていればよい。

URL 埋め込みの利点は、長い入力さえ厭わなければ、通常のクライアントでもアノテーション付きコンテンツを取得することが可能な点である。しかし、URL 埋め込みの方法ではアノテーション付きコンテンツに含まれるすべての URL をアノテーション付きのコンテンツを指す URL に変更する必要がある。このために、「元々の Web 空間」と「アノテーション付きの Web 空間」の間の移動の際には、ユーザはモードを変えて reload するという操作が必要になり、アノテーションヘッダに比べると、2つの空間の移動が自由ではない。

分離プロキシの利点は、“Netscape Proxy Automatic Configuration”のようなプロキシの自動設定の機能を持つブラウザならば、クライアントを変更せずに簡単にアノテーション付きのコンテンツにアク

セスできる点である。しかし、この方法では検索キーワードのような付加情報を送ることは諦めなければならない。

4. 実 装

ここでは、前章の設計に基づき Squid Internet Object Cache²²⁾を改造してアノテーションサーバを実装した。このサーバは、「アノテーションヘッダ」「URL 埋め込み」「分離プロキシ」の3つの手法に対応している。ユーザ情報の交換には HTTP Proxy-Authentication²⁰⁾の機能を用いている。本サーバは SPARC Ultra2(SunOS 2.5.1), SGI O2(IRIX 6.3), PC/AT 互換機(FreeBSD)で稼働している。

アノテーションサーバでは、前章で述べたように、コンテンツの種類ごとに別のアノテーション処理モジュールを用意する。各アノテーション処理モジュールは、アノテーション埋め込みの関数である *annoMerge()* と、アノテーションの解釈および登録の関数である *annoSave()* を持つ。これらの関数は、アノテーションサーバのコンテンツ転送のルーチンから呼ばれる。たとえば、HTML データに対してのアノテーションの埋め込み処理は、アンカー() タグを付加することであり、「HTML データ中からタグを付ける文字列を探し、タグを付ける文字列の前後にタグを埋める」という2つの手順からなる処理ルーチンである。また、MPEG データに対してリンク情報を埋める場合は、Cmew/M システムでは SYSTEM ストリームの PRIVATE パケットにリンク情報を埋め込むので、「PRIVATE パケットの先頭を探し、リンク情報を PRIVATE パケットに書き込む」という処理ルーチンである。

ここで、アノテーションサーバの処理の流れを説明する。クライアントの發したコンテンツの獲得要求(GET リクエスト)に対して、アノテーションサーバはプロキシと似た次のような処理を行う。

- (1) クライアントのリクエストを受ける。
- (2) リクエストから URL とアノテーション属性を取り出す。
- (3) URL のコンテンツを持つ Web サーバ(または、他のプロキシ)にコンテンツを要求する。
- (4) コンテンツが獲得できたら、獲得したコンテンツをアノテーション埋め込みルーチン(*annoMerge()*)に渡す。
- (5) アノテーション埋め込みの終わったコンテンツをクライアントに渡す。

プロキシサーバは通常、Web サーバからのデータを

もちろん、他の2つの手法もユーザ情報の伝達に HTTP Proxy-authentication を利用することができる。

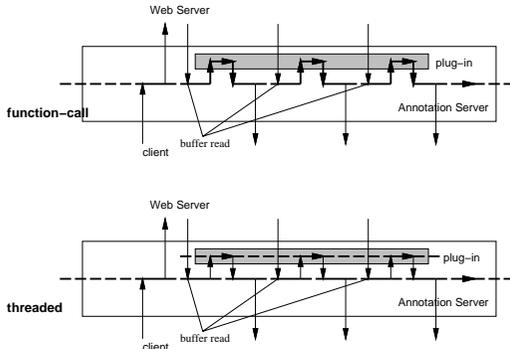


図6 プラグイン処理のマルチスレッド化
Fig. 6 Multi-threaded plug-in processing.

クライアントに中継する際に、ファイルのすべてをいったんダウンロードしてから転送するのではなく、プロキシ内に持つバッファ(4KB程度)が一杯になったらクライアントに転送を開始することで遅延を小さくおさえている。アノテーションサーバでもこれと同様に、クライアントの要求およびサーバからバッファに読み込んで、その中身に応じてアノテーションの埋め込みおよび転送の処理を行う。このためアノテーション埋め込みルーチンにはこのバッファ長を単位としてデータが渡される。ところが、通常、この受渡しの際のバッファの切れ目はデータ処理の切れ目とは合致しない。そこで、プラグインには再呼び出しを考慮しつつ未処理のデータを次の呼び出しまで保存する処理を実装しなくてはならない。

アノテーションサーバの初期の実装では、アノテーションサーバ-プラグイン間でこれらの未処理データや、Webから読み出したコンテンツ、埋め込むアノテーションといったデータを保持する特定のデータ構造をもって情報の受渡しを行っていた。そのために、プラグインを作成するプログラマは、未処理データをこのデータ構造に埋めておいて、次回に繰り越すという処理をプログラムする必要があった。そこで、アノテーションサーバの内部を、プロキシとしての転送処理を受け持つスレッドとプラグインを処理するスレッドに分離することで、プラグインのプログラムが複雑になることを回避した(図6)。ここでは、プラグインを構成するためのツールキットを作成し、プロキシとのデータのやりとりやコンテキスト切替えの処理をツールキットの処理ルーチン内部に隠すことで、プラグインのプログラムが転送の切れ目に関する同期を意識する必要がなくなった。

アノテーションの保存については、HTTPのPUTメソッドを使っている。アノテーションヘッダ方式と

表1 アノテーションサーバのオーバーヘッド
Table 1 Overhead of the annotation server.

アノテーション	null	1	5	10
オーバーヘッド (ms)	30	337	384	457

URL埋め込み方式の場合、アノテーション属性の付いたPUTリクエストの<BODY>と</BODY>には含まれる部分をアノテーションの中身としてプラグインに渡し、プラグインが適当な形で保存する。

5. 評価

ここでは評価として、アノテーションサーバのアノテーション埋め込みプラグインでの処理オーバーヘッド、および、アノテーションサーバを経由した場合の転送レートの比較を示す。

表1にサーバ側プラグインによる処理のオーバーヘッドを示す。これは、Sun Ultra2(Ultra SPARC 200MHz, Solaris 2.5.1)で稼働しているサーバで、テキストに対するアノテーションプラグインを用いて、150KB程度のテキストファイルにアノテーションを挿入した場合のオーバーヘッドを調べたものである。ここでは、アノテーションの挿入回数1~10個に変えた場合のオーバーヘッドを調べた。nullは何もアノテーションを挿入しないで、単にプラグインを呼び出すだけのオーバーヘッドである。この結果をみると、アノテーションの挿入のプラグイン実行によるオーバーヘッドは、数百msかかっているが、元データのインターネットからのダウンロードの速度・遅延を勘案すると十分満足できるものであると考えている。

次に、アノテーションサーバを経由するデータ転送のオーバーヘッドを測定した。ここでは、入力そのまま出力するだけのnullプラグインを作成し、localhostのWebサーバ上の5Kバイト、20Kバイト、1Mバイト、5Mバイトのファイルの転送レートを100回測定しその平均を求めた。ここで、プラグインに入力から1回に読み込むデータ量の大きさを変えたものを作成して、プラグインの単位処理量の違いによるオーバーヘッドの違いをみた。測定結果を表2に示す。表中でSquidとあるのはSquid Proxyサーバ経由の転送レートで、Bufferとあるのはマルチスレッド化する前のアノテーションサーバの転送レートである。プラグイン入出力での単位処理量としては1バイト、8バイト、64バイト、512バイト、1Kバイト、4Kバイトの6種類を測定した。

ここでは、小さいサイズのファイル転送に関しては、結果に大きければつきがでているが、これは多重化の

表2 アノテーションサーバ経由の転送レート
Table 2 Transfer rate through the annotation server.

単位：KByte/sec (Squid との比)

size (byte)	他実装		処理単位					
	Squid	Buffer	1 Byte	8 Byte	64 Byte	512 Byte	1 KByte	4 KByte
5 K	516.2 (1.00)	454.1 (0.88)	317.8 (0.62)	94.6 (0.18)	94.7 (0.18)	94.9 (0.18)	87.2 (0.16)	244.8 (0.47)
20 K	1560.1 (1.00)	1240.8 (0.79)	658.9 (0.42)	1131.8 (0.72)	1386.8 (0.88)	1461.2 (0.93)	1003.7 (0.64)	1006.3 (0.64)
1 M	538.7 (1.00)	469.5 (0.87)	354.4 (0.65)	450.2 (0.83)	486.1 (0.90)	485.7 (0.90)	487.4 (0.90)	490.1 (0.91)
5 M	366.6 (1.00)	354.8 (0.97)	301.2 (0.82)	345.5 (0.94)	358.3 (0.97)	368.1 (1.00)	371.4 (1.01)	370.8 (1.01)

スケジューリングやプロトコルオーバーヘッド等の影響が全処理量に比して大きいと考えられる。ばらつきの大きい部分を除いた全体的な傾向としては、プラグインでの入出力処理単位が小さい程転送効率が低下している。処理単位の小さいところで20～30%、大きいところで10%程度の効率低下になっている。しかし、実際のインターネット経由のデータ転送を考えると、この程度の転送効率の差はユーザにとってはあまり問題にはならない。

6. 考 察

本研究では、著者らが開発してきた Cmew システムに対して、情報を付加する機能および付加情報を共有する機能を提供するアノテーションサーバシステムを実現した。このアノテーションサーバは、複数のメディアに対してのリンクやコメントといった情報を直接かつ動的に付加し、それらの付加情報の共有を可能にしている。ここで実現した Web アノテーションの機能は、直観的には本に対する書き込みのようなものである。既存の Web 環境には大量のマルチメディア情報が存在するが、それらに対してメモを書き込んだり、各自でハイパーリンクを直接増やしたり、余計なリンクを消したりといった形で、既存の情報に手を入れることはできない。これに対して、Web に対する書き込みの機能を提供すれば、既存の情報をより使いやすい形に変えることができ、また興味の似通った人や専門の知識を持つ人が付加した情報やリンク等の制御情報を共有することで、自分に必要なより多くの情報や専門性の高い情報を獲得することが期待できる。

この機能を位置的に分散して存在するメンバー間での議論等の場において利用すると、議論の対象となる様々なメディアコンテンツに対して直接情報を付加し、それを他のメンバーと共有するという形態で、Cmew (Web) というシステムを介して情報の共有や議論が可能になる。他の応用としては、作成に多大な労力の

かかるリンク付きストリームメディアコンテンツの作成過程での利用が考えられる。アノテーションサーバを使うことで、複数の作業者がリンク付加工具を用いて動画像や音に対して付加したリンク情報を動的に集約することが可能になり、作業の効率化が期待できる。

従来からも Web に対するアノテーションシステムの事例はあるが、テキストが中心であり、複数のメディアに対応したものはない。また、サーバ側にプラグインを用意して挙動を変えるものとして、Apache Web Server 等の Web サーバで提供されている SSI (Server Side Includes) があげられるが、本システムはプロキシのフィルタとしてプラグインの機能を提供している。今後、様々なメディアの登場やブラウザの多様化に対して、本アノテーションサーバの拡張性の機能は重要な技術である。また、今後はユーザ個人の嗜好を反映した多様な挙動が求められると考えられるので、その面でも本システムは主要な技術になる。

本システムでは、プロキシがアノテーションを保持する形態をとった。5章に示したように、処理オーバーヘッドの点ではこの方式のオーバーヘッドは十分に小さく問題がないと考えている。アノテーション埋め込みをサーバ側やクライアント側で行う方式に対する、本プロキシ方式の利点としては次のようなものが考えられる。

まず、アノテーションをプロキシが保持することにより、どの Web サーバのデータに対してもアノテーションを付加することが可能となる。サーバ側にアノテーションを保持する場合には、アノテーション機能を提供していないサーバに対してはアノテーション機能を利用することができない。また、本プロキシ方式では、アノテーションサーバが稼働しアノテーション埋め込み機能が提供されている間は、つねに複数ユーザ間のアノテーション共有の機能が提供される。一方、クライアント側にアノテーションを保持した場合には、

他ユーザのクライアントが停止している間はそのユーザの付けたアノテーションは利用できなくなる。

プロキシ方式の他の利点としては、埋め込みの多様かつ複雑な機能をプロキシの機能として提供することで、クライアント側に特殊なプログラムを配布する必要がなくなるという点があげられる。クライアント側で埋め込むようにすると、ブラウザの種類と埋め込みの機能の組合せで多数の埋め込みソフトウェアを提供する必要が生じる。さらに、ユーザ管理の観点に立つと、サーバ側で保持する方式は管理する対象が広くなり非現実的である。特に本研究の目指している、同一グループ等のある程度限られたユーザ間でのアノテーション共有に対しては、プロキシ方式が最も有効な解法であると考えられる。

今回はアノテーションの登録に関しては PUT メソッドを用い、アノテーションの表現形式も Cmew システム独自のものであるが、WEBDAV 等での規格化が進めば、それに追従することは新しいプラグインを用意するだけで済むので容易である。また、現在の実装では元のオリジナルコンテンツが更新されて整合性が崩れた場合、変更された元のデコンテンツについてアノテーションは意味がないという立場にたつて、アノテーションは埋め込まない。しかし、元コンテンツの変更が小規模の場合には、変更差分を考慮して可能な限りアノテーション埋め込みをする方法も考えられるので、この点については今後の検討および改良が必要であると考えている。

7. ま と め

本稿では、Web コンテンツに情報を付加しそれを共有するシステムである Cmew/U の設計と実装について述べた。マルチメディアコンテンツに対する Web アノテーションを考慮したシステムは従来にないものである。今回実装したアノテーションサーバは、アノテーション埋め込みや保存の機能がプラグインとして実現されており、メディアの種類やクライアントの表示機能等に応じて柔軟に拡張することが可能である。このアノテーションサーバの性能については、インターネットからの Web データの転送を考慮にいと、十分に満足できる転送速度が実現されている。

謝辞 本研究の初期の段階で、数々のご支援と貴重な意見をいただきました、電気通信大学尾内理紀夫教授に深く感謝いたします。また、本研究を進めるうえでご支援いただきました、NTT 未来ネット研究所ネットワークインテリジェンス研究部小柳恵一部長に感謝いたします。

参 考 文 献

- 1) 高田敏弘, 青柳滋己, 佐藤孝治, 廣津登志夫, 尾内理紀夫: WWW における連続メディアへのリンク情報の附与, 画像電子学会第 9 回メディア統合技術研究会 (1997).
- 2) 高田敏弘, 青柳滋己, 佐藤孝治, 廣津登志夫, 尾内理紀夫: Cmew: 連続メディアと WWW の統合, *Proc. Japan WWW Conference '97* (1997).
- 3) Takada, T., Sato, K., Aoyagi, S., Hirotsu, T. and Sugawara, T.: Cmew: Integrating Continuous Media with the Web, *Proc. 3rd Annual Multimedia Technology and Applications Conference (MTAC '98)*, Anaheim, CA, U.S.A., pp.136-140 (1998).
- 4) 高田敏弘, 有村浩一, 尾内理紀夫: 応答性/対話性を重視したビデオマークアップエディタ, pp.15-22, 近代科学社 (1997).
- 5) 青柳滋己, 高田敏弘, 佐藤孝治, 廣津登志夫, 菅原俊治, 尾内理紀夫: 音データへのリンク情報の埋め込み法, コンピュータソフトウェア, Vol.16, No.6, pp.13-23 (1999).
- 6) Aoyagi, S., Takada, T., Sato, K., Hirotsu, T. and Sugawara, T.: Cmew/A: An Access Method for Audio Data with Link Information, *Proc. 6th International Conference on Distributed Multimedia Systems (DMS '99)*, Aizu, Japan, pp.26-33 (1999).
- 7) 佐藤孝治, 高田敏弘, 青柳滋己, 廣津登志夫, 菅原俊治, 尾内理紀夫: 連続メディアと WWW の統合システム Cmew におけるシナリオ制御機構の実現, コンピュータソフトウェア, Vol.16, No.3, pp.47-56 (1999).
- 8) Sato, K., Takada, T., Aoyagi, S., Hirotsu, T. and Sugawara, T.: Dynamic Multimedia Integration with the WWW, *Proc. 1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 1999)*, Victoria, B.C., Canada, pp.448-451 (1999).
- 9) 廣津登志夫, 高田敏弘, 青柳滋己, 佐藤孝治, 尾内理紀夫: マルチメディアデータのための Plug-in Annotation Server の設計と実装, *Proc. Japan WWW Conference '97* (1997).
- 10) 廣津登志夫, 高田敏弘, 青柳滋己, 佐藤孝治, 菅原俊治, 尾内理紀夫: Multimedia Annotation Server のための Plug-in Toolkit の設計と実装, 第 1 回プログラミングおよび応用のシステムに関するワークショップ (SPA '98) (1998).
- 11) Hirotsu, T., Takada, T., Aoyagi, S., Sato, K. and Sugawara, T.: Cmew/U — A Multimedia Web Annotation Sharing System, *Proc. IEEE Region 10 Conference (TENCON '99)*, Cheju, Korea, pp.356-359 (1999).

- 12) *Mosaic User's Guide: Annotations*. <http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/help-on-annotate-win.html>
- 13) *Group Annotations in NCSA Mosaic*. <http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/group-annotations.html>
- 14) *CoNote HomePage*. <http://wheat.tc.cornell.edu/pub/davis/annotation.html>
- 15) Röscheisen, M., Mogensen, C. and Winograd, T.: Beyond Browsing: Shared Comments, SOAPs, Trails, and On-line Communities, *Proc. 3rd International World-Wide Web Conference*, Darmstadt, Germany (1995).
- 16) *ComMenter HomePage*. <http://walrus.stanford.edu/diglib/pub/reports/WWW95/>.
- 17) WEBDAV Working Group: Requirements for Distributed Authoring and Versioning on the World Wide Web (1997). Internet-Draft (draft-ietf-webdav-requirements-02.txt)
- 18) *WEBDAV HomePage*. <http://www.ics.uci.edu/~ejw/authoring/>
- 19) Berners-Lee, T., Fielding, R.T. and Nielsen, H.F.: Hypertext Transfer Protocol — HTTP/1.0 (1996). RFC1945.
- 20) Fielding, R.T., Gettys, J., Mogul, J.C., Nielsen, H.F., Masinter, L., Leach, P.J. and Berners-Lee, T.: Hypertext Transfer Protocol — HTTP/1.1 (1999). RFC2616.
- 21) Franks, J., Hallam-Baker, P.M., Hostetler, J.L., Lawrence, S.D., Leach, P.J., Luotonen, A. and Stewart, L.C.: HTTP Authentication: Basic and Digest Access Authentication (1999). RFC2617.
- 22) *SQUID Internet Object Cache HomePage*. <http://www.squid-cache.org/>

(平成 12 年 10 月 30 日受付)

(平成 13 年 9 月 12 日採録)



廣津登志夫

1967 年生。1995 年慶應義塾大学大学院理工学研究科計算機科学専攻博士課程修了。同年、日本電信電話株式会社入社。基礎研究所を経て現在、未来ねっと研究所に所属。入社以来、マルチメディアシステム、分散システム、OS、ネットワーク等の研究に従事。博士(工学)。日本ソフトウェア科学会、ACM、IEEE-CS 各会員。



高田 敏弘

1962 年生。1986 年東京工業大学理学部情報科学科卒業。1988 年同大学大学院理工学研究科情報科学専攻修士課程修了。同年、日本電信電話株式会社入社。基礎研究所を経て現在、未来ねっと研究所に所属。その間 1994 年スタンフォード大学訪問研究員。入社以来、並列オブジェクト指向計算、大規模分散システム、インターネット情報システムの多言語化、多相情報システム等の研究に従事。



青柳 滋己(正会員)

1965 年生。1988 年東京工業大学理学部情報科学科卒業。1990 年 3 月同大学大学院理工学研究科情報科学専攻修士課程修了。同年、日本電信電話株式会社入社。現在、同社未来ねっと研究所に所属。複合メディア情報処理の研究に従事。



佐藤 孝治(正会員)

1967 年生。1989 年慶應義塾大学理工学部数理科学科卒業。1991 年同大学大学院理工学研究科計算機科学専攻修士課程修了。同年、日本電信電話株式会社入社。現在、同社未来ねっと研究所に所属。分散システム、マルチメディアシステム等に興味を持つ。



菅原 俊治(正会員)

1982 年早稲田大学大学院理工学研究科(数学専攻)修士課程修了。同年、日本電信電話公社入社(武蔵野電気通信研究所基礎研究部)。以来、知識表現、学習、分散人工知能、マルチエージェントシステム、インターネット等の研究に従事。1992~1993 年、マサチューセッツ大学アムハースト校客員研究員。現、NTT 未来ねっと研究所主幹研究員。博士(工学)。日本ソフトウェア科学会、IEEE、ACM 各会員。