WebDAV システムのセキュアな設定・運用に関する調査

調査報告書

(現状調査編)

平成 15 年 4 月

情報処理振興事業協会

セキュリティセンター

1. 本編の概要

本調査では、セキュアな WebDAV システムについて、その利用を促進するために設定方法を総括する調査を行うとともに、ソフトウエア開発が必要と考えられるアクセス権管理機能を中心としたフィジビリティスタディ等を行い、今後の技術開発の必要性についての検討を示す。[1]

「現状調査編」においては、現在の WebDAV 関連仕様および実装状況に関する調査結果を示す。以下の項目について調査を行った。

- WebDAV 関連仕様

  WebDAV の基本機能と拡張機能について、発行された RFC と Internet-Draft を基に機能やプロトコルに関し調査を行った。基本機能については RFC2518 で定義された機能を、拡張機能については RFC3253 で定義されたバージョンコントロール機能および Internet-Draft で定義されているアクセスコントロール、リソース検索の機能を調査の対象とした。また WebDAV を活用する上で必要となるユーザ認証についても調査を行った。

- WebDAV を利用可能なソフトウエア

  現在利用可能な WebDAV のサーバおよびクライアントに関し、それぞれ代表的な実装を対象として、機能および WebDAV 実装レベルに関し調査を行った。

- 類似するオーサリングツール

  WebDAV と目的および機能の面で類似するコンテンツオーサリングツールとして CVS を取り上げ、両者の比較を行った。

[1] 本調査は、調査主体である情報処理振興事業協会セキュリティセンター（IPA/ISEC）が、株式会社 SRA 先端技術研究所に委託して平成 14 年度に実施した。

2. WebDAV 関連仕様

この章では、WebDAV の基本機能の仕様および拡張機能の仕様に関して記述する。

2.1. 概要

2.1.1. WebDAV の仕様とは

WebDAV は HTTP/1.1 [2] に分散環境におけるコンテンツの作成･編集に関する機能を追加し、コンテンツの作成から、公開、閲覧までの一連の作業を HTTP 上で行うことを可能にするプロトコルである。

WebDAV プロトコルの開発と標準化は IETF におけるワーキンググループ活動を中心に進められている。

2.1.2. 調査の対象

次の点に関して調査を行った。

- WebDAV の基本仕様
- WebDAV の拡張仕様
  - リソース検索機能
  - アクセスコントロール機能
  - バージョン管理機能
- 類似オーサリングツール

[2] RFC2616 “Hypertext Transfer Protocol -- HTTP/1.1” http://www.ietf.org/rfc/rfc2616.txt

2.1.3. 調査手法

以下の手法により調査を行った。

（1） 標準化された仕様の調査

主に以下のインターネット標準を参照した。

- RFC2291 “Requirements for a Distributed Authoring and Versioning Protocol for the World Wide Web” [3]
- RFC2518 “HTTP Extensions for Distributed Authoring – WEBDAV” [4]
- RFC3253 “Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)” [5]
- RFC2617 “HTTP Authentication: Basic and Digest Access Authentication” [6]

（2） 標準化作業中の仕様の調査

主に以下のインターネットドラフトを参照した。

- draft-reschke-webdav-search-03
- draft-ietf-webdav-acl-09

（3） ソフトウエアの調査

主に各ソフトウエアのドキュメントや Web ページに公開された情報を参照した。また動作時のログ等の解析による調査も行った。

（4） 関連技術・ツールの調査

関連技術やツールに関する公開ドキュメントや Web ページを参照し調査を行った。

その他、WebDAV Resources[7]を中心とするインターネット公開情報を参考とした。本編の末尾の参考文献に有用な情報源を示す。

---

[3] RFC2291 "Requirements for a Distributed Authoring and Versioning Protocol for the World Wide Web" http://www.ietf.org/rfc/rfc2291.txt

[4] RFC2518 "HTTP Extensions for Distributed Authoring -- WEBDAV" http://www.ietf.org/rfc/rfc2518.txt

[5] RFC3253 "Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)" http://www.ietf.org/rfc/rfc3253.txt

[6] RFC2617 "HTTP Authentication: Basic and Digest Access Authentication" http://www.ietf.org/rfc/rfc2617.txt

[7] WebDAV Resources http://www.webdav.org/

2.2. WebDAV 仕様

ここでは WebDAV の基本仕様について、主に RFC2291、RFC2518 の記述を基に述べる。

2.2.1. 標準化の現状

WebDAV に関する仕様の策定は、IETF において進められている。すなわち、IETF において組織されたワーキンググループにおける議論をベースに、進捗に応じドラフト仕様が発行され、さらに議論と修正を行った末に RFC が発行される。WebDAV の標準化については、IETF の Applications エリア内に設立された webdav ワーキンググループが仕様策定にあたっている[8]。

また、Web に関する標準化については、IETF の他に W3C（World Wide Web Consortium）がアプリケーションレイヤにおいて大きな役割を果たしており、プロトコルレイヤを担当する IETF と協力して標準化活動を行っている。

Webdav ワーキンググループにおいて、2003 年 2 月時点で RFC として標準化された仕様を以下に示す。

- RFC2291 “ “ Requirements for a Distributed Authoring and Versioning Protocol for the World Wide Web ”
- RFC2518 “HTTP Extensions for Distributed Authoring – WEBDAV ”
- RFC3253 “Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)”

2.2.2. WebDAV の概要

(a) 背景

Web 関連技術の発展と普及により、ユーザがインターネット上のコンテンツにアクセスすることは容易になったが、その一方でコンテンツを作成する上ではさまざまな問題が残されている。Web の基幹プロトコルである HTTP にはリモートからのコンテンツ作成を可能とする機能が含まれているが、これらは基礎的な機能しか持たず、Web サーバソフトウエアにおいても十

[8] IETF WEBDAV Working Group　http://www.ics.uci.edu/~ejw/authoring/

分なサポートは行われていなかったと言える。このため各種のオーサリングソフトウエアではHTTP に独自の拡張を施すものや、ftp など HTTP 以外のプロトコルを利用するものなどが多くあった。

WebDAV は、このようなコンテンツ作成に関する問題を解決するために、HTTP の拡張仕様として提唱されたプロトコルである。

(b) 目標

WebDAV の目標を以下に示す。

- リモートからのコンテンツ作成を可能にする
- コンテンツのバージョン管理を可能にする
- アクセスコントロール、検索などコンテンツに対する多様なアクセス手段を実現する

(c) 特徴

WebDAV の特徴を以下に挙げる。

■ HTTP の拡張仕様である

WebDAV は HTTP/1.1 の拡張仕様であり、HTTP/1.1 の枠組み内にいくつかの新しいメソッド、ヘッダ等を追加することにより実現されている。既存のメソッド、および URI のメカニズムに互換性の問題を起こすような変更は加えられておらず、HTTP/1.1 との互換性を維持している。

■ シンプルである

HTTP を拡張するものであるため、プロトコルはクライアントからの要求に対してサーバが応答を返すシンプルなものであり、実装が容易である。また、通信はすべて HTTP 上で行われるため、ファイヤーウォール等の既存のネットワーク基盤との親和性は高く、運用も容易である。

■ OS を問わない

一般に、ファイルを取り扱うプログラムの機能は、ファイルの持つ属性やアクセス管理などの点で OS のファイルシステムが提供する機能に大きく影響されるが、WebDAV はこれらの点を抽象化しており、原理的には OS を問わず利用可能である。

2.2.3. HTTP/1.1 より拡張された概念

WebDAV では、HTTP/1.1 に対していくつかの概念を追加した。ここでは拡張された概念として、リソース、コレクション、プロパティ、ロックについて詳細を示す。

(a) リソース

リソースとは、ネットワーク上に存在するデータ、サービスなど、URI[9]で識別され、WebDAV において操作の対象となるものである。つまり HTTP/1.1 でアクセス可能なすべての情報はリソースである。

(b) コレクション

コレクションとは、他のリソースを含むリソースであり、リソースをまとめて扱うものとして定義される。コレクションはファイルシステムにおけるディレクトリに相当する概念であるが、両者は決して同一のものではない。

URI は階層構造を持つので、ルートを除くあらゆる URI はその URI を含むコレクションのメンバーとみなされる。例えば http://www.example.com/foo/ がリソースであり、http://www.example.com/foo/bar がリソースである時、前者はコレクションであり、後者は前者のメンバーである。

(c) プロパティ

プロパティとは、リソースに対して与えられた、そのリソースに関する情報であり、リソースに関する一種のメタデータである。プロパティは名前と値の組で表される。以下の 2 種類が存在する。

■ **ライブプロパティ**（Live Property）

サーバによって与えられるプロパティ。ファイルサイズのように、実行時にサーバによって与えられるものである。リソースが実在するオブジェクトである場合は、そのリソースにはライブプロパティが必ず存在すると考えられる。

ライブプロパティには、サーバによって維持されるものと、クライアントによって維持されサーバにより文法チェックが行われるものの二種類が存在する。

---

[9] RFC 2396 “Uniform Resource Identifiers” http://www.ietf.org/rfc/rfc2396.txt

■ **デッドプロパティ**（Dead Property）

クライアントによって与えられるプロパティ。サーバはクライアントの要求によりデッドプロパティを格納し、また要求に応じて返答するだけであり、内容に関しては関知しない。

プロパティは WebDAV の拡張プロトコルにおいても、必要なメタデータを格納する手段として幅広く用いられる。

(d) ロック

リソースへの直列化されたアクセスを保証するため、WebDAV ではリソースのロック機構が追加されている。

■ **ロックとは**

一般的にロックは排他制御の為に用いられる。

ロックが必要な理由を説明する。WebDAV により同一のドキュメントを編集しようとするクライアント A と B があるとする。編集作業は人間が行うため、ある程度時間がかかる。A、B 双方が同時にドキュメントを取得して編集をはじめ、まず B が作業を終了してドキュメントを上書きしたとする。しかしその後 A が作業を終了してドキュメントを上書きしてしまうと、B の編集結果はすべて失われてしまう。このような問題は「更新の消失問題」として知られる。

これを防ぐためにはリソースに対するアクセスが、A のドキュメント取得、編集、上書き、B のドキュメント取得、編集、上書き、と直列化される必要がある。ロックを用いると先の例のアクセスは、A によるロック、ドキュメント取得、編集、上書き、ロックの開放、B によるロック、ドキュメント取得、編集、上書き、ロックの開放と直列化され、「更新の消失問題」は回避される。

■ **排他ロックと共有ロック**

WebDAV で使用可能なロックは排他ロック（exclusive lock）と共有ロック（shared lock）の 2 種類である。

排他ロックは最も基本的なロック形態であり、ロックをかけたクライアント以外のクライアントが同じロックをかけることができない。ロックに失敗したクライアントがロックの開放を待つことで「更新の消失問題」は防止される。

現実の分散システム運用においては排他ロックの使用には難しい点がある。例えばロックを取得したクライアントがアプリケーションの不正終了により正しくロックを開放できなかった場合、他のクライアントはタイムアウトによるロックの自動解除を待つか、あるいは管理権限を持つシステム管理者等によるロックの強制解除を行う必要が生じる。

これに対して共有ロックでは、同じリソースに対して他のクライアントもロックをかけることが可能である。ロックの強度は排他ロックに劣るが、ロックの存在を示すことにより、アプリケーション間の自発的な協調を促すことは可能である。共有ロックを「紳士的に」取り扱うアプリケーションは、アクセスしたリソースが共有ロックされていた場合には、何らかの手段でロックの所有者にアクセスの意思を示そうとする。

■ ロックトークン

ロックを行うとサーバによりロックトークンが発行され、レスポンスのプロパティに含まれる形でクライアントに返される。ロックトークンはロックの識別子であり、そのサーバの全てのリソースに対して、永久に一意に定まるものでなければならない。一般にこのような用途には ISO-11578 で定義される UUID[10]メカニズムが用いられるが、一意性を満たす限りは任意のトークン作成メカニズムを使用できる。クライアントに返されたロックトークンはアンロックの際に使用される。

■ ロックの実装

ロックはファイルシステムと密接に関係するため、RFC はロックの実装を必須とはせず、また実装する際に必須なロックの種類も定義していない。そのためクライアントは、まず HTTP/1.1 の OPTIONS メソッドにより、該当するリソースにおいてサーバがロックをサポートしているかどうかを調査し、次に supportedlock プロパティを取得して使用可能なロックの種類を取得する必要がある。

■ WebDAV におけるロックの制約

WebDAV においてロックを用いる際には、WebDAV 自身の持つさまざまな制約について考慮する必要がある。

まずロックはファイルシステムと密接な関係にあるが、WebDAV はプロトコルであり OS とは独立している点がある。つまりプロトコルにおいてロックの詳細な動作を規定することは難しく、またロックのサポートを必須とすることも困難である。

次に HTTP との関係がある。WebDAV は HTTP/1.1 の拡張であるが、HTTP/1.1 にはロックの概念は存在しない。つまり WebDAV でロックをサポートしたとしても、HTTP/1.1 のみに準拠するクライアントに対しては、ロックの効果は及ばない。必然的に WebDAV におけるロックはクライアント間の協調が必要なアドバイザリロックとならざるを得ない。

WebDAV クライアントは、リソースの更新を行う際には可能な限りロックを利用することが良いと考えられる。これは仕様上の要件ではないが、大規模分散環境において必要な紳士的な振る舞いのひとつと考えられる。また HTTP/1.1 クライアントも、リソースの更新時に If-Match ヘッダを利用することで、他者の更新を消滅させることを防止できる。

---

[10] Universal Unique Identifier

2.2.4. プロトコルの詳細

ここでは WebDAV のプロトコルの詳細について述べる。

(a) メッセージボディの記述法

WebDAV では、リクエストとレスポンス双方のメッセージボディの記述に XML を使用する。使用されるメッセージは全て整形式（Well-formed）の XML 文章でなければならない。サーバとクライアントのどちらにおいても、不正な XML 文章によるメッセージは評価することなく破棄することが要求されている。

(b) HTTP/1.1 より拡張されたメソッド

WebDAV において、HTTP/1.1 に拡張されたメソッドを以下に示す。

（1） PROPFIND

指定されたリソースのプロパティを取得するメソッドである。リクエストのボディで要求するプロパティを XML で記述する。

例として URI http://webdav.example.com/webdav/file1.pdf に対して PROPFIND メソッドを発行する場合を示す。以下はリクエストである。

```
PROPFIND /webdav/file1.pdf HTTP/1.1
Host: webdav.example.com
Depth: 0
Content-Type: text/xml; charset="utf-8"
Content-Length: 120

<?xml version="1.0" encoding="utf-8" ?>
<D:propfind xmlns:D="DAV:">
<D:prop>
<D:creationdate/>
</D:prop>
</D:propfind>
```

ここでは creationdate プロパティを指定して、ファイルの作成日時をリクエストしている。これに対する応答を次に示す。

```
HTTP/1.1 207 Multi-Status
Date: Mon, 10 Feb 2003 11:18:26 GMT
Server: Apache/2.0.43 (Unix) mod_ssl/2.0.43 OpenSSL/0.9.6b DAV/2
Content-Length: 348
Content-Type: text/xml; charset="utf-8"

<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
<D:response xmlns:lp1="DAV:" xmlns:lp2="http://apache.org/dav/props/">
<D:href>/webdav/file1.pdf</D:href>
<D:propstat>
<D:prop>
<lp1:creationdate>2003-02-10T10:42:20Z</lp1:creationdate>
</D:prop>
<D:status>HTTP/1.1 200 OK</D:status>
</D:propstat>
</D:response>
</D:multistatus>
```

このように、ファイルの作成日時が返される。

この例では単一のリソースに対してアクセスしているが、コレクションに対するアクセスの場合では、Depth ヘッダによって動作が変化する。Depth として 0 を指定するとコレクション自体への、1 もしくは infinity を指定するとコレクションのメンバーも対象に含めた動作となる。

（2） PROPPATCH

PROPPATCH は、URI で指定されたリソースに対してプロパティの設定・削除を行う。リクエストのボディに XML 文章で命令を記述する。複数の命令を同時にリクエスト可能であり、命令は全て実行されるか、全て拒否されるかのいずれかである。つまり一つでも実行不能の命令が含まれていればリクエスト全体が失敗となる。

（3） MKCOL

リクエストされた URI に新しいコレクションを作成する。コレクションが既に存在していた場合は失敗となる。作成されたコレクションは、ルートである場合を除き、親コレクションのメンバーに追加される。

MKCOL の実行例を以下に示す。まずリクエストの例を示す。

```
MKCOL /webdav/testcol/ HTTP/1.1
Host: webdav.example.com
```

この例では、コレクション http://webdav.example.com/webdav/testcol/ の作成を要求している。正常に作成された場合の応答を以下に示す。

```
HTTP/1.1 201 Created
```

サーバによってはこの後にメッセージボディとして HTML が返されるが、ここでは省略している。

（4） COPY

COPY メソッドは、Request-URI ヘッダで指定されたリソースを Destination ヘッダで指定された URL へコピーする。リソースのデッドプロパティはコピーされ、ライブプロパティは同様の働きを持つプロパティが作成される。コレクションに対する COPY は Depth によって動作が指定される。Depth を 0 以外にすることで、再帰的なコピーが実行される。

（5） MOVE

MOVE メソッドは、Request-URI ヘッダで指定されたリソースを Destination ヘッダで指定された URL へコピーし、移動元のリソースを削除する。リソースのデッドプロパティはコピーされ、ライブプロパティは同様の働きを持つプロパティが作成される。一連の処理はアトミックに行われる。MOVE メソッドは、Depth ヘッダに Infinity が指定されたものとして再帰的に実行される。

（6） LOCK

Request-URI で指定したリソースおよびそのプロパティをロックする。リクエストの Depth ヘッダは 0 もしくは Infinity の値のみを取ることが可能で、それぞれ指定したリソー

スのみ、もしくはリソースに含まれるすべてのメンバーに対して再帰的にロックが適用されることを意味する。

ロックの種類は、メッセージボディ中の lockscope エレメントで指定する。

ロックが成功すると、lockdiscovery プロパティがレスポンスボディに含まれて返される。このプロパティにはロックトークンが含まれている。

LOCK メソッドの実行例を示す。以下の例は www.example.com 上の/webdab/file1.pdf に対して、排他ロックを行う要求である。

```
LOCK /webdav/file1.pdf HTTP/1.1
Host: www.example.com
Timeout: Infinite, Second-4100000000
Content-Type: text/xml; charset="utf-8"
Content-Length: 249

<?xml version="1.0" encoding="utf-8" ?>
<D:lockinfo xmlns:D='DAV:'>
 <D:lockscope><D:exclusive/></D:lockscope>
 <D:locktype><D:write/></D:locktype>
 <D:owner>

<D:href>http://www.example.com/~webdav/contact.html</D:href>
 </D:owner>
</D:lockinfo>
```

この要求に対する応答を以下に示す。

```
HTTP/1.1 200 OK
Date: Sat, 15 Feb 2003 14:55:23 GMT
Server: Apache/2.0.43 (Unix) mod_ssl/2.0.43 OpenSSL/0.9.6b DAV/2
Lock-Token: <opaquelocktoken:05924cbe-b6b6-0310-b47a-8a2f7e50823a>
Content-Length: 487
Content-Type: text/xml; charset="utf-8"

<?xml version="1.0" encoding="utf-8"?>
<D:prop xmlns:D="DAV:">
<D:lockdiscovery>
```

```
<D:activelock>
<D:locktype><D:write/></D:locktype>
<D:lockscope><D:exclusive/></D:lockscope>
<D:depth>infinity</D:depth>
<ns0:owner xmlns:ns0="DAV:">
   <ns0:href>http://www.example.com/~webdav/contact.html</ns0:href>
 </ns0:owner><D:timeout>Infinite</D:timeout>
<D:locktoken>
<D:href>opaquelocktoken:05924cbe-b6b6-0310-b47a-8a2f7e50823a</D:href>
</D:locktoken>
</D:activelock>
</D:lockdiscovery>
</D:prop>
```

ロックは成功し、ロックトークンとして<opaquelocktoken:05924cbe-b6b6-0310-b47a8a2f7e50823a>が返されている。

（7） UNLOCK

Request-URI で指定したリソースにかけられた、Lock-Token ヘッダで示されたロックトークンで識別されるロックを解除する。LOCK を実装する場合は必ず UNLOCK も実装しなければならない。

（8） 拡張された既存メソッド（DELETE、PUT）

リソースとコレクションの概念の導入に伴い、HTTP/1.1 に存在するメソッド DELETE と PUT に対しては拡張が行われた。

- DELETE

  リソースに対するリクエストではリソースが所属する全ての親コレクションからの削除を行い、コレクションに対する DELETE では指定されたコレクションとそのメンバーであるリソースを全て削除する。

- PUT

  リソースに対する PUT 時にプロパティを書き換えることが拡張された。コレクションに対する PUT としては MKCOL が定義されている。

(c) HTTP ヘッダ

WebDAV において拡張された HTTP ヘッダを以下に示す。

（1） DAV

DAV ヘッダがサポートされていることと、DAV クラスを示す。

（2） Depth

メソッドの適用範囲を示す。0 の場合はリクエスト URI のみに、1 の場合は対象リソースとその直接のメンバーであるリソースに、Infinity の場合はリソースと全ての子に適用される。

（3） Destination

COPY メソッドと MOVE メソッドにおいてリソースの移動先の URI を示す。

（4） If

HTTP/1.1 の If-Match ヘッダと同様にステートに応じた処理を行う。

（5） Lock-Token

ロックトークンを示す。

（6） Overwrite

COPY メソッドと MOVE メソッドにおいて、移動先に上書きするかを示す。

（7） Status-URI

STATUS CODE として 102 Processing が返される際に処理中の URI を示す。

（8） Timeout

ロックのタイムアウト時間を示す。

(d) STATUS CODE

WebDAV において拡張されたステータスコードを以下に示す。

（1） 102 Processing

リクエストは受理されたが処理に一定以上の時間がかかる場合に返される。

（2） 207 Multi-Status

複数の応答が含まれる場合に返される。

（3） 422 Unprocessable Entity

リクエストボディの XML は整形式であるがセマンティクスがおかしい場合などに返される。

（4） 423 Locked

リソースはロックされている。

（5） 424 Failed Dependency

依存関係にある他のメソッドが失敗した為に実行できない場合に返される。

（6） 507 Insufficient Storage

実行に必要なサーバの記憶領域が不足している場合に返される。

(e) プロパティ

RFC2518 で定義された WebDAV のプロパティを以下に示す。これら基本的なプロパティ以外にも、実装により独自に拡張されたプロパティや、WebDAV 拡張機能において定義されたプロパティが多数存在する。

（1） creationdate

リソースの作成日時。

（2） displayname

リソースの表示名。

（3） getcontentlanguage

GET メソッドを発行した際に付加される Content-Language ヘッダの値。

（4） getcontentlength

GET メソッドを発行した際に付加される Content-Length ヘッダの値。

（5） getcontenttype

GET メソッドを発行した際に付加される Content-Type ヘッダの値。

（6） getetag

GET メソッドを発行した際に付加される ETag ヘッダの値。

（7） getlastmodified

GET メソッドを発行した際に付加される Last-Modified ヘッダの値。

（8） lockdiscovery

リソースにかけられているアクティブなロックのオーナー、タイプ、タイムアウト、ロックトークンなどの情報を示す。

（9） resourcetype

リソースの種類を示す。

（１０） source

リクエスト URI の処理前のソースの URI。

（１１） supportedlock

利用可能なロックの種類。

2.2.5. DAV 準拠クラス

WebDAV 仕様の実装にあたっては、2 種類のクラスから準拠レベルを選択することが許されている。WebDAV クライアントは OPTIONS メソッドによりアクセスするサーバの準拠レベルを知ることができる。

WebDAV 仕様を実装する際には、まず WebDAV は HTTP/1.1 の拡張仕様として提案されている為、RFC2616 で定義されている HTTP/1.1 の仕様に準拠しなければならない。その上で、実装する WebDAV 準拠レベルを選択する。

RFC2518 で定義されている WebDAV 準拠クラスは次の 2 種類である。

（1） クラス 1

クラス 1 は、RFC2518 中において「MUST」で示された要件を満たさなければならない。また、サーバは OPTIONS メソッドの応答に DAV ヘッダとして”1”を含めなければならない。

（2） クラス 2

クラス 2 は、Class 1 に準拠した上で、LOCK メソッド、および supportedlock と lockdiscovery の両プロパティ、Time-Out と Lock-Token の両ヘッダを実装しなければならない。また、サーバは OPTIONS メソッドの応答に DAV ヘッダとして”1”と”2”を含めなければならない。

2.2.6. 動作の実際

WebDAV プロトコル仕様は RFC として示されているが、その記述は最低限従うべき基本的な仕様を示すものである。アプリケーションが実際どのようにプロトコルを使用しているか等の実現方法の詳細に関しては各実装に委ねられている。動作の実際に関して公開されている情報は多くはないが、インターネットコミュニティに実装経験が集まればベストプラクティスとしてメモが公開されることもある。

ここでは代表的な WebDAV サーバとクライアントの組み合わせにおいて、いくつかの操作を行った際に発行されるメソッドについて述べる。

(a) WebDAV フォルダの閲覧

HTTP/1.1 に対応したウエブブラウザはディレクトリに対するアクセスに GET メソッドを発行する。一般的なサーバ実装においては、ディレクトリに対して設定するデフォルトのドキュメント（index.html など）を検索し、もしこれらが存在しない場合にリストアップが許可されている場合にはディレクトリに含まれるファイルをリストアップした html データを作成してクライアントへ返す。ディレクトリに含まれるファイルへのアクセスは、html 化されたファイルリストからハイパーリンクをたどることにより実現される。

WebDAV クライアントでは、DAV が有効かどうかやサポートされているクラスを調べるためにまず OPTIONS メソッドを発行する。DAV が有効な場合、サーバからのレスポンスヘッダに Dav ヘッダが含まれている。

Dav が有効なことが分かると、クライアントはファイルやディレクトリの一覧を取得するために PROPFIND メソッドを発行する。例えばオープンソースの WebDAV クライアントである cadaver の場合、PROPFIND メソッドにより主に次のプロパティを取得している。

- getcontentlength（ファイルサイズの表示に使用する）
- getlastmodified（更新日時の表示に使用する）
- displayname（ファイル名の表示に使用する）
- executable（Apache の拡張プロパティであり、実行可能であることを示す）

クライアントは、PROPFIND メソッドの応答を解析してこれらのプロパティを取り出し、WebDAV フォルダの内容を表示する。

(b) ファイルの取得

WebDAV クライアントがリモートにあるファイルを取得する際には、まずファイルの URI を知る必要がある。ウエブブラウザの場合は、URL はハイパーリンクや人間の入力などにより既

知である場合が多いが、WebDAV の場合は PROPFIND メソッドの結果が使用される。PROPFIND メソッドの発行の詳細については WebDAV フォルダの閲覧と同様である。

ファイルの取得には GET メソッドが使用される。これは HTTP/1.1 によるアクセスと同様である。

実装上の検討が必要な点として、対話的な環境においては、PROPFIND メソッドを発行してファイル一覧を取得してから実際にファイルを取得するまでに、ユーザの操作待ちなどによりある程度の時間がかかる点を挙げられる。主にパフォーマンス上の理由から PROPFIND の結果を長時間キャッシュして再利用する実装がある。

RFC2518 では第三者によるアクセス等による状況の変化が考えられるので PROPFIND の結果はキャッシュされるべきではないとしている。実際にキャッシュを行っても、読み取りの際には大きな問題は発生しないと考えられるが、書き込みの際には上書きによるデータの喪失を回避するために、HEAD メソッドの再実行などによりキャッシュとの整合性を確認する必要がある。

(c) ファイルの作成

ファイルを作成する際に、対話的な WebDAV クライアントでは、まず対象となる URL に対して HTTP/1.1 の HEAD メソッドが発行される。HEAD メソッドは実際にはコンテンツの取得を行わない GET メソッドのように位置づけ可能であり、GET アクセスをシミュレートできる。

HEAD メソッドの応答により対象 URL に既にリソースが存在することが判明した場合は、強制上書きや処理のキャンセルなどの選択肢をユーザへ提示する処理を行うべきである。多くの対話的な WebDAV クライアントはこのような処理を行っている。

HEAD メソッドがレスポンスとして 404 Not Found を返しリソースが存在しないことが分かった場合は、PUT メソッドを発行しファイルをアップロードする。

(d) ディレクトリの作成

対話的な WebDAV クライアントがディレクトリを作成する際には、ファイルの作成と同様に、まず HEAD メソッドを発行し、対象となる URL に既存のコレクションが存在するかどうかを調査する。もし既存のコレクションが存在する場合は処理をキャンセルする。存在しない場合は、MKCOL メソッドを発行してコレクションを作成する。

GUI を持ついくつかの環境では、ディレクトリを新規作成する際に、まず規定の初期ディレクトリ名で作成し、その後ユーザにディレクトリ名の入力を促すものがある。例えば Windows や OS X では「新しいフォルダ」等の初期名でフォルダを作成する。このようなクライアントは HEAD メソッドに引き続いて初期ディレクトリ名を含んだ URL で MKCOL メソッドを発行して初期コレクションを作成する。その後、ユーザに実際のフォルダ名の入力を要求し、MOVE メソッドを発行してコレクションの移動を行う。

2.2.7. WebDAV における認証

RFC2518 においては新たな認証メカニズムは定義されていない。WebDAV は HTTP/1.1 を拡張したプロトコルであるため、WebDAV の認証は HTTP/1.1 に準拠する。

HTTP/1.1 における認証は RFC2617[11]に定義されている。RFC2617 で定義される認証方式には次の 2 種類がある。

（1） Basic 認証
（2） Digest 認証

これらの詳細については 2.6 ユーザとグループの」にて述べる。

2.2.8. セキュリティ

ここでは、WebDAV のセキュリティについて述べる。

(a) HTTP/1.1、XML との関係性

これまでにも述べてきたように、WebDAV は HTTP/1.1 の拡張として実現されている。そのため HTTP/1.1 に関するセキュリティ上の問題は WebDAV にもあてはまる。

HTTP/1.1 のセキュリティ上の問題としては、暗号化されていない通信路を用いるためデータが盗聴される問題、Basic 認証のパスワードが容易に盗聴され得る問題などがある。これらの問題についてはそれぞれの項目で後述する。

また、プロトコル中で XML を用いているため、XML に関するセキュリティ上の諸問題についても WebDAV にあてはまることに注意しなければならない。XML の外部参照の機能を用いて DTD や外部参照実体にアクセスする場合、それらが書き換えられた際に問題が発生し、セキュリティ上のリスクと成り得ることが知られている。例えば DTD の書き換えにより WebDAV のリクエストボディに含まれる XML が不正になった場合、仕様によりリクエスト全体が無効となり、サービス拒否攻撃が成立する。

---

[11] RFC2617 ”HTTP Authentication: Basic and Digest Access Authentication” http://www.ietf.org/rfc/rfc2617.txt

(b) 認証の必要性

コンテンツの編集を可能にしている関係上、WebDAV においては、アクセス制限だけでなく悪意の第三者によるコンテンツの汚染を防ぐために認証機能が必要となる。またサービス拒否攻撃を防止するためのロックにも認証機能が必要である。

(c) 認証の暗号化

認証に関連して特に注意が必要な点として、HTTP/1.1 は素の状態では通信が暗号化されていない点が挙げられる。暗号化されていない通信路では第三者による通信内容の傍受が可能である。WebDAV に認証機能を持たせる際には、HTTP/1.1 における Basic 認証がパスワードを平文で送信することに特に注意しなければならない。WebDAV 仕様ではセキュアでない通信路では Basic 認証を利用してはならないとしている。

Basic 認証に代わる認証方式として RFC2069 で定義される Digest 認証は、クリアテキストの送信を行わずに認証を行う方式であり、Basic 認証に比べ、より安全な認証を行える。WebDAV 仕様では全ての WebDAV アプリケーションは Digest 認証をサポートしなければならないと定めている。

(d) プロパティの操作

現状の多くの Web システムの運用においては、外部に公開しないディレクトリについては空の（またはアクセス不可である旨を示す）インデックスファイルを作成し、ディレクトリに含まれるファイルの存在を隠蔽する運用手法が広く用いられている。しかしながら WebDAV では、PROPFIND メソッドを用いてコレクションに対してプロパティ一覧の取得を再帰的に行うことで、コレクションに含まれるリソースの存在を知ることができる。このことにより上記のようなファイルの隠蔽手法は無効化されてしまう。

またロックの際には、コンタクト情報としてロックのオーナーに関する情報がプロパティとして格納される。これらの情報はアプリケーション間の自発的な協調のために使用される個人情報（アプリケーションに関するユーザ ID、メールアドレス、インスタントメッセージ ID など）を含んでおり、これらのプロパティへのアクセスは制限する必要がある。またクライアントもコンタクト情報の送信およびその内容についてはユーザがコントロール可能でなければならない。

ロックのコンタクト情報以外にも、プロパティにはリソースに関するさまざまな情報が格納され、特にリソースの作成者や編集者の情報が含まれる場合が多いと考えられる。これらプライバシーにかかわるプロパティへのアクセスは慎重にコントロールされるべきである。

(e) サービス妨害攻撃

WebDAV システムは、本質的にサービス妨害攻撃に対して弱点となり得る点を多く持つ。

WebDAV はリモートからのリソース作成をサポートするため、非常に大きなリソースを作成することでディスク空間を浪費させる攻撃を受ける可能性がある。また、プロパティの操作についても、巨大なコレクションに対して再帰的なプロパティ操作命令を発行することで処理能力を浪費させられる可能性がある。

これらの攻撃は、WebDAV アプリケーションにとどまらず、同一ホストで動作する全てのサービスに影響を与えると考えられるため、WebDAV システムの運用にあたっては特に慎重に考慮する必要がある。

(f) ロックトークンに関する問題点

WebDAV では、ロックトークンに一意性を保証するために UUID を用いる。しかしこれにより、サーバが作成したロックトークンを取得することで、サーバの IEEE 802 アドレスをリモートのネットワークから取得される可能性がある。これにより、サーバもしくはサーバのネットワークインタフェースのメーカーが特定され、既知の弱点を利用した攻撃をされ易くなるなどセキュリティ上のリスクが発生する可能性がある。

(g) Source Link に関する問題点

一般に Web サーバでは CGI などのスクリプトのソースへのアクセスを制限している。また通常スクリプトは read only である。しかし、WebDAV ではリソースの Source プロパティを取得することで、スクリプトのソースへアクセス可能な形の URI を得られる潜在的な可能性が存在する。書き込みが可能な WebDAV では、これにより保護されていたはずのスクリプトのソースが保護されなくなる可能性がある。スクリプトの編集にも WebDAV を用いる場合は十分な注意が必要である。

2.2.9. 国際化

ここでは、RFC2518 で述べられている内容を元に、WebDAV 仕様の国際化に関わる点について述べる。

WebDAV においては、メソッドや XML タグなどではない、人間が読むことのできる（human-readable な）テキストはプロパティの値か、エラーメッセージである。どちらも XML

でエンコードされているため、キャラクターセットやエンコードは XML のメカニズムによって指定される。つまり XML の”xml:lang”属性によって言語が指定される。XML の仕様により、WebDAV プロトコルを扱う XML プロセッサはエンコードとして最低限 UTF-8 を取り扱える必要がある。UTF-8 以外のサポートは実装ごとのオプションである。

全ての WebDAV アプリケーションは、キャラクターセットのタグ付け、エンコーディング、言語タグの機能をそれぞれサポートしなければならない。

WebDAV 仕様に登場する名前については次の 3 種類が存在する。

- プロトコルエレメント名
- XML エレメント名
- プロパティ名

プロトコルエレメント名の文字コードには、HTTP/1.1 と同様に USASCII を用いる。プロトコルエレメント名は処理系内に閉じた名前であり通常は人間には見えないため、国際化する必要性は存在せず、よって WebDAV 仕様においても国際化の対象とされていない。

XML エレメント名は UTF-8 でエンコードされた英語名が使用されるが、プロトコルエレメント名同様にユーザには見えないため、複数のキャラクターセットを用いる必要は無い。よって XML エレメント名も WebDAV 仕様においては国際化の対象とされていない。

プロパティ名については、プロパティは URI として定義されている。アプリケーションによっては、プロパティ表示時に URI を人間に読みやすい形式にマッピングする機能を有することも考えられるが、このような機能はアプリケーションにあらかじめ用意されたプロパティセットに対してのみ有効に機能する。

以上より、WebDAV アプリケーションは人間により可読な形式のプロパティ名を使用することが推奨される。

WebDAV のエラーメッセージは、HTTP/1.1 のステータスコードを踏襲している。国際化されたアプリケーションの多くは、内部でステータスコードを詳細な記述に変換する機能を既に持っている。このため WebDAV 仕様はエラーメッセージの国際化および詳細化を対象としない。

2.3. WebDAV のリソース検索機能

ここでは、WebDAV のリソース検索機能（以下 DASL と省略する）について述べる。

2.3.1. 概要

DASL（DAV Searching and Locating）とは、WebDAV について、コンテンツの内容やプロパティ値などを利用したリソースの検索機能を拡張するものである。HTTP と WebDAV の組み合わせによりクライアント側における検索は可能であるが、たとえば全文検索を行う際に全てのコンテンツを取得する必要があるなど、ネットワーク帯域の使用効率および処理効率は悪い。それに対して、DASL はサーバ側における検索機能を提供するため、ネットワーク帯域の使用を節約するだけでなく、サーバ側で過去の検索結果のキャッシュやコンテンツのインデックス付け等を行うことにより検索処理自体の効率化をはかることが可能である。

DASL の使用モデルは以下のように記述できる。

（1） クライアントが DAV:basicsearch プロパティを用いたクエリを作成する。
（2） クライアントが検索を行うリソースに対して SEARCH メソッドを発行する。メッセージボディにはクエリが含まれる。
（3） サーバが検索を行う。
（4） サーバが検索結果を返す。メッセージボディは PROPFIND の応答と同様である。

DASL 仕様の開発は IETF の WebDAV ワーキンググループ内の DASL ワーキンググループにて行われている[12]。調査時点においては、プロトコルはドラフト段階であり、IETF から Internet-Draft として発行されている。調査時点の最新のドラフトは 2003 年 2 月に発行された draft-reschke-webdav-search-03 である。ドラフトは主に以下の内容から構成されている。

（1） 追加されるメソッド、ヘッダ、プロパティの定義
（2） クエリの記述方法

以下にその詳細について述べる。

[12] http://www.webdav.org/dasl/

2.3.2. 追加されたメソッド

(a) SEARCH

DASL では、HTTP メソッドとして SEARCH が追加された。SEARCH メソッドはクライアントより発行され、これを受けてサーバで検索が行われる。DASL における検索は、HTTP を使用している関係で、一回のクエリに対して一回の応答が行われる単純なものである（ただし応答内容に複数の検索結果が含まれることはある）。

■ リクエスト URI

SEARCH メソッドを発行する対象となる URI には、検索の「仲介者」となる URI を指定する。この URI は特別なものではなく、全ての HTTP リソースを用いることが可能である。WebDAV リソースである必要は無い。SEARCH メソッドでは、クエリ文法において特に定義されている場合を除いて、仲介者である URI と検索範囲（スコープ）との間には何の関係も定義されていない。

■ リクエストボディ

リクエストのボディとしてクエリを記述する。プロトコルは XML で記述されたクエリを処理できなければならないが、XML で記述されていなければならないとは規定されていない。XML 以外の文法で記述されたクエリを処理することも可能である。

クエリは、DAV:searchrequest エレメントを含まなければならない。

■ 応答

SEARCH メソッドの応答を以下に示す。

検索が成功した場合の応答は、PROPFIND メソッドの応答と同様になる。応答中には一つ以上の DAV:response エレメントが存在し、それぞれの response は検索にマッチしたリソースの URI を持つ。それ以上の応答の詳細は、各クエリ文法において定義される必要がある。ここではこれ以上詳細には扱わない。

検索が失敗した場合は、サーバはエラーを示すステータスコードと共に、検索がどのスコープで失敗したかを示すエレメントを返さなければならない。

(b) OPTIONS

DASL を利用するクライアントは、各仲介者 URI でどのようなクエリ文法が利用可能であるかを知る必要がある。そのため DASL では OPTIONS メソッドに対して、利用可能なメソッドとして SEARCH を返すことと、応答中に DASL ヘッダを含めることが追加されている。

2.3.3. 追加された HTTP ヘッダ

（1） DASL

要求されたリソースで利用可能なクエリ文法を示すヘッダである。文法は URI で識別される。例を以下に示す。

DASL: <http://www.example.com/syntax1>
DASL: <DAV:basicsearch>

2.3.4. 追加されたプロパティ

（1） DAV:supported-query-grammar-set

指定されたリソースで利用可能なクエリ文法のセットを示す。

2.3.5. クエリスキーマの発見

OPTIONS メソッドの応答に含まれる DASL ヘッダは利用可能なクエリ文法を示すが、クライアントにおいてクエリを作成するためには、例えば検索対象として指定可能なプロパティなどの情報が必要となり、DASL ヘッダから得られる情報だけでは不十分なこともある。このような実際の検索に必要な情報を取得することを、クエリスキーマの発見と呼ぶ。

DASL においてクエリスキーマを発見するためには、クエリ文法とスコープを指定した DAV:query-schema-discovery エレメントを含む SEARCH メソッドを使用する。

以下にクエリスキーマ発見のための SEARCH メソッドの例を示す。

```
SEARCH /search HTTP/1.1
Host: www.example.com

<query-schema-discovery xmlns="DAV:">
 <basicsearch>
```

```
  <from>
   <scope>
    <href>http://www.example.com/news/</href>
    <depth>infinity</depth>
   </scope>
  </from>
 </basicsearch>
</query-schema-discovery>
```

この例では、www.example.com/search に対して、/news/以下全体に対して DAV:basicsearch 文法を用いた検索要求を発行する場合のスキーマを取得している。

なお、DAV:basicsearch 文法の詳細については、Internet-Draft 中において定義されている。

2.3.6. DASL のセキュリティ

DASL は他の WebDAV 拡張仕様と同様に、HTTP および WebDAV のセキュリティ要件の影響を受ける。また XML の仕様に含まれる外部参照によるセキュリティ上のリスクも存在する。

実装時には DASL で検索を行う際に PROPFIND など他のメソッドで許可されていないリソースに対する検索が行われないようにしなければならない。

サービス妨害攻撃に関する考慮も必要である。膨大な検索処理時間を必要とするクエリや、結果が膨大な量にのぼるクエリを与えられた際の動作については特に検討を要する。

2.4. WebDAV のアクセスコントロール機能

ここでは、WebDAV のアクセスコントロール機能について述べる。

2.4.1. 概要

WebDAV のアクセスコントロールプロトコルは WebDAV 仕様に対する拡張仕様として開発が進められている（以下これを WebDAV ACL 仕様と呼ぶ）。

分散環境におけるコンテンツ作成を目的とする WebDAV では、リソースに対して不特定のアクセス主体（ユーザー）が閲覧、または更新・修正作業を行うことが予測されるが、その際にはどのユーザはどのリソースに対するアクセスが可能であり、またどのリソースに対しては不可能であるといったアクセスコントロールを行うことが要求される。基本となる WebDAV 仕様にはそのような機能が存在しないため、WebDAV サーバが保有しているリソースおよびプロパティに対するこのようなアクセスコントロールを可能にすることを目標に拡張仕様が検討されている。

仕様の開発は、IETF の WebDAV ワーキンググループのサブワーキンググループとして設置された WebDAV Access Control Protocol ワーキンググループにおいて行われている[13]。調査時点において、WebDAV ACL 仕様はドラフト段階であり、IETF から Internet-Draft として発行されている。最新ドラフトは 2002 年 7 月に発行された draft-ietf-webdav-acl-09 である。

Internet-Draft で定義された仕様は以下の内容を含んでいる。

（1） PRINCIPAL（アクセス主体）の定義
（2） アクセス権の定義
（3） アクセスコントロールのプロトコル

これらの仕様はまだドラフト段階のものであり、今後変更されることも予想される。メソッドの実行に必要なアクセス権の定義など、重要な点が引き続き議論中で仕様化は行われていない。

以下にその詳細について述べる。

[13] WebDAV Access Control Protocol　http://www.webdav.org/acl/

2.4.2. PRINCIPAL の定義

PRINCIPAL とは、“ アクセスしているのは誰か ” を定義するものである。ユーザやある種のソフトウエア（エージェント等）が PRINCIPAL に相当する。仕様においては、PRINCIPAL は URI によって表現される。PRINCIPAL は複数の URI によって表現することが可能であり、その時そのうちの一つの URI は http（または https）の URL でなければならない。

しかし PRINCIPAL が複数の URI で表現可能であることは、アクセス権の設定の際に混乱を招きかねない。提案仕様ではクライアントがアクセス権を設定する際に、単一の PRINCIPAL に統一すべきであるとしているが、柔軟性がある反面問題も残していると考えられる。

PRINCIPAL へのアクセスは、PROPFIND と PROPPATCH メソッドによってプロパティを取得・設定することで行われる。よって WebDAV ACL を提供するサーバは最低限 PROPFIND メソッドを実装することが必要である。

PRINCIPAL はグループであることがある。グループ PRINCIPAL はメンバーである PRINCIPAL を代表するものと定義される。このグループとメンバーの関係は再帰的であり、あるユーザがグループ A のメンバーであり、かつグループ A がグループ B のメンバーであるときは、そのユーザはグループ B のメンバーでもある。

2.4.3. アクセス権の定義

リソースに対して発行されたメソッドの動作は、アクセス権（PRIVILEGES）によってアクセスの可否がコントロールされなければならない。また各メソッドの実行にどのようなアクセス権が必要とされるかも、仕様で定義される必要がある。リソースに対するアクセス権を持たない PRINCIPAL は、定義された特別な場合（アクセス権が全許可、疑似などの場合）を除きすべてサーバにより拒否されなくてはならない。

アクセス権は複数のアクセス権を集合させた形で表現することも可能であり、サーバは自由に集合アクセス権を実装することができる。ただし DAV:write に DAV:read を含んではならないなどの制限は存在する。

WebDAV ACL 仕様に示されるアクセス権は DAV:名前空間を用いているが、もし実装により独自のアクセス権を追加する場合は別の名前空間を用いなければならない。

WebDAV ACL 仕様で提案されているアクセス権を以下に示す。

（1） DAV:read

GET および PROPFIND メソッドに対してリソースの情報を取得することを許可する。OPTIONS メソッドを制御することも可能である。

（2） DAV:write

PUT や PROPPATCH のようなメソッドに対し、リソースの更新とロック、およびデッドプロパティの設定を許可する。

（3） DAV:write-properties

PROPPATCH のようなメソッドに対し、デッドプロパティの設定を許可する。

（4） DAV:write-content

PUT、DELETE に対し、リソースの更新を許可する。

（5） DAV:unlock

LOCK の所有者でない PRINCIPAL に対し UNLOCK を許可する。主に管理目的に使用される。

（6） DAV:read-acl

PROPFIND メソッドによる DAV:acl プロパティの取得を許可する。

（7） DAV:read-current-user-privilege-set

PROPFIND メソッドによる DAV:current-user-privilege-set プロパティの取得を許可する。

（8） DAV:write-acl

ACL メソッドによる DAV:acl プロパティの設定を許可する。

（9） DAV:all

対象となるリソースに適用可能な全ての PRIVILEGE を表す。

ただし、2003 年 2 月時点においては、これらのアクセス権の各メソッドの適用の詳細については決定されていない。WebDAV ACL 仕様は WebDAV に関連する全ての仕様に対応する必要があるため、拡張仕様が追加されるごとに、それぞれに対する PRIVILEGE が策定されなければならない。

2.4.4. ACL と ACE

ACE（Access Control Element）は、ある PRINCIPAL に対するアクセス権を示したセットである。ACE を記述する主な XML エレメントを以下に示す。

（1） Principal

ACE が適用される PRINCIPAL を示す。主なエレメントを示す。

URI	PRINCIPAL の URI
All	全てのユーザ
authenticated	全ての認証済みユーザ
unauthenticated	全ての非認証ユーザ

（2） grant、deny

適用されるアクセス権を示す。

grant	許可されるアクセス権
deny	拒否されるアクセス権

ACL（Access Control List）は、あるリソースに対する ACE の集合である。ACL は ACE を子に持つ XML エレメントとして記述される。

ACL の記述例を以下に示す。

```
<D:acl>
 <D:ace>
 <D:principal><D:href>http://www.example.com/acl/gourps/webadmins</D:href></D:principal>
  <D:grant><D:privilege><D:write/></Dprivilege></D:grant>
 </D:ace>
 <D:ace>
  <D:principal><D:all/></D:principal>
  <D:grant><D:privilege><D:read/></D:privilege></D:grant>
 </D:ace>
</D:acl>
```

この例では、http://www.example.com/acl/gourps/webadmins で識別されるユーザグループに対して write 権限を、それ以外のユーザに対しては read 権限を与える ACL を記述している。

ACL に含まれる各 ACE の適用方法は、ACL Semantics により定義される。

（1） DAV:ace-combination

複数の ACE がどのように組み合わされるかを指定する。

first-match	アクセス権が認められない ACE が登場した段階で失敗となる
all-grant-before-any-deny	ACE が拒否か、最終的に許可されない場合に失敗となる

（2） DAV:ace-ordering

ACE の評価順を指定する。

deny-before-grant	grant より先に deny の評価をする

（3） DAV:allowed-ace

どのような ACE が ACL 中で許可されるかを指定する。

principal-only-one-ace	principal は一つの ACE にのみ指定可能
grant-only	grant のみ指定可能
no-invert	invert エレメントは使用不可

2.4.5. プロパティ

WebDAV ACL 仕様において、WebDAV 仕様から拡張されたプロパティを示す。拡張されたプロパティは、大きく PRINCIPAL に対するものと、アクセスコントロールに対するものに分けられる。

(a) PRINCIPAL のプロパティ

WebDAV ACL 仕様においては、PRINCIPAL はリソースであると定義され、URI により識別される。リソースであるため、PRINCIPAL はプロパティを持つ。

PRINCIPAL は RFC2518 で定義された次のプロパティを持たなければならない。

（1） DAV:displayname

（2） DAV:resourcetype

値として DAV:principal エレメントを持たなくてはならない。

また、RFC2518 に加えて、以下のプロパティを定義する。

（1） DAV:alternate-URI-set

PRINCIPAL に関する追加の情報を持つ。値は URI で表される。クライアントに対して、PRINCIPAL の詳細を明らかにすることが目的である。例として LDAP 形式のアドレスなどが考えられる。

（2） DAV:principal-URL

定義により PRINCIPAL は複数の URI を持てるが、一意の識別を保証する為にそのうちの一つをプライマリ URI として用いなければならない。本プロパティは ACL 要求に設定され、プライマリ URI を値に持つ。

（3） DAV:group-member-set

グループに含まれる PRINCIPAL を持つ。

（4） DAV:group-membership

グループの識別子を持つ。

(b) アクセスコントロールプロパティ

アクセスコントロールプロパティは、リソースに対して設定されたそのリソースへのアクセスの許可・不許可等を示すプロパティである。

以下に主なプロパティを示す。

（1） DAV:owner

どの PRINCIPAL がリソースの所有者であるかを示す。

（2） DAV:supported-privilege-set

リソースに対して設定可能なアクセス権を示す。アクセス権はプロパティであるが、人間に理解可能にするために description エレメントを用いてその概要を記述する。

（3） DAV:current-user-privilege-set

アクセスした認証済み HTTP ユーザに許可されているアクセス権のセットを示す。

（4） DAV:acl

リソースに対して ACL を設定するプロパティである。

（5） DAV:acl-semantics

DAV:acl プロパティで示された ACL の適用ルールを示す。詳細については 2.4.4ACL と ACE」を参照のこと。

2.4.6. メソッド

(a) RFC2518 から拡張されたメソッド

WebDAV ACL 仕様においては、RFC2518 に存在する以下の HTTP メソッドに対して拡張が提案されている。

（1） OPTIONS

OPTIONS では、WebDAV ACL のサポートを示すヘッダが追加される。ACL をサポートするサーバは DAV ヘッダに “ access-control ” を含めなければならない。

（2） MOVE、COPY、DELETE、LOCK

実行時の DAV:acl プロパティの扱いが追加された。

(b) 新規追加されたメソッド

WebDAV ACL 仕様においては、RFC2518 に対しては以下のメソッドが新たに追加提案されている。

（1） ACL

ACL メソッドはリソースに対して設定された ACL を変更するメソッドである。ACL に含まれる保護されていない ACE が変更可能である。設定された ACL は PROPFIND メソッドにより取得される。

（2） REPORT

REPORT メソッドは RFC3253 で拡張された、リソースに関する情報を取得するメソッドである。PROPFIND よりも複合的な処理を行う。

2.4.7. WebDAV ACL における認証

WebDAV ACL 仕様に基づきアクセスコントロールを実現するためには、認証メカニズムが必須となる。WebDAV ACL 使用における認証は基本仕様と同様に HTTP/1.1 の認証スキームによって行われる。すなわち、Basic 認証と Digest 認証が用いられる。

WebDAV ACL 仕様においては、リソースに対するアクセスは誰が（PRINCIPAL）どのような権利を持つか（PRIVILEGES）によって示される。PRINCIPAL は新しく定義された URI 名前空間に存在する。しかし HTTP の認証はアクセス対象であるリソースに対して設定されているので、ACL では PRINCIPAL と HTTP 認証のユーザとをマップさせる必要がある。クライアントにおけるマップを容易にするために認証された PRINCIPAL を返すレスポンスヘッダを追加することについての議論と検討が調査時点でも続けられている。この詳細については今後の Internet-Draft または RFC の発行により明らかになると期待される。

また、WebDAV ACL 仕様ではユーザグループを定義し、ユーザグループをアクセス主体とした ACE を定義することが可能である。

2.5. WebDAV のバージョン管理機能

ここでは、WebDAV のバージョン管理機能について述べる。

2.5.1. 概要

WebDAV バージョン管理機能は、WebDAV に対してリソースのバージョン管理を可能にする拡張機能である。

WebDAV におけるバージョン管理とは、ユーザに対して以下の操作を提供することを表す。

- リソースをバージョン管理下に置く
- 更新されたリソースを自動的に新しいバージョンとして登録する設定
- リソースの任意のバージョンへのアクセス

その他に、Web リソースに対して並行的にバージョンを管理する機能を提供するアドバンストバージョン管理機能と呼ばれる追加機能も提供する。

WebDAV バージョン管理機能は IETF の WebDAV ワーキンググループ内の Delta-V ワーキンググループ[14]において標準化が現在も進められている。

バージョン管理機能は 2002 年 3 月に RFC3253 として発行された[15]。調査時点において、WebDAV の拡張機能の中で唯一 RFC となったものである。現在、修正作業が進められている。修正版は RFC3253 を置き換える新しい RFC として発行されるものと考えられる。

2.5.2. バージョン管理の概要

バージョン管理とは、テキストファイルやソースコードなどのさまざまなリソースに対し、その変更履歴を管理することを意味する。OS でサポートされる場合を除けば、通常ファイルシステム上のファイルはバージョン管理されていない。つまり、変更前のファイルも変更後のファイ

[14] IETF DELTA-V Working Group Home Page http://www.webdav.org/deltav/
[15] Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning) http://www.ietf.org/rfc/rfc3253.txt

ルも同じファイルであり、意図的にバックアップを取るなどして差分を取らない限りは変更内容も明らかにはならない。

これに対してバージョン管理システムでは、ファイルに対する変更作業を監視し、変更に応じてその時点のファイルにバージョンとして ID を付加していく。また、あるバージョンのファイルそのもの、またはバージョン間の差分を保存しておくことにより、ID を指定した特定のバージョンへのアクセスを実現する。これにより変更の取り消し、複数人による同一ファイルの編集の管理等が容易になる。

CVS に代表されるバージョニングシステムは、一般に以下のような機能を提供する。

- 既存バージョンの保護
- 既存バージョンのロック
- 既存バージョンの取り出し
- 新バージョンへの ID の付与と管理
- 新バージョンの書き込み
- ロックの開放
- 既存バージョンの保護の開放

WebDAV においては、上のいくつかの操作をまとめてひとつの操作として提供している。ファイルの取り出しに関連する操作は非可分（アトミックオペレーション）なチェックアウト（check out）、新しいバージョンの格納および後処理は、同様に非可分なチェックイン（check in）機能として提供されている。

2.5.3. WebDAV のバージョン管理モデル

WebDAV の拡張として規定されるバージョン管理機能で採用されているバージョンモデルでは、あるひとつのバージョンから次のバージョンを生成する連続したバージョン生成の他に、ひとつのバージョンから複数のバージョンを生成するフォーク（fork）あるいはブランチ（branching）、異なる複数のバージョンからひとつのバージョンを生成するマージ（merging）が提供されている。

バージョンの関係はグラフとして表現できる。このグラフをバージョングラフと呼ぶ。WebDAV のバージョン管理機構は、バージョングラフを管理することで実現される。バージョングラフに対して提供される主な機能を以下に示す。

- バージョングラフ全体の参照
- バージョングラフの特定のノード（バージョン）の参照
- クライアントへのバージョン管理されているリソースと管理されていないリソースの判別機能

■ バージョンのヒストリーデータの管理、取り出し
■ バージョングラフへのナビゲーション
■ バージョングラフのトポロジーの管理
■ 特定バージョンの固定化
■ クライアントからのバージョン ID の指定
■ バージョン ID の一意性の管理
■ クライアントからのバージョンツリーのクエリの発行

2.5.4. 基本バージョン管理機能

WebDAV サーバで提供されるバージョン管理機能は、つぎの３つのパッケージを提供しなければならない。

(1) Core-Versioning Package: version-control
(2) Basic-Server-Workspace Package: version-control, workspace, version-history, checkout
(3) Basic-Client-Workspace Package: version-control, working-resource, update, label

(a) Core-versioning Package

Core-versioning Package は、バージョン管理機能を提供するクライアントおよび提供しないクライアントに対して提供されるもので、リニアにバージョン番号が更新されるような、単純に連続したバージョンサービスを提供するパッケージである。バージョン管理機能を提供するクライアントは、バージョン管理されている一つ前の版のリソースおよびそのプロパティに対するアクセスが可能となる。

(b) Basic-Server-Workspace Package

Basic-Server-Workspace Package は、並行的なバージョン管理機能であるアドバンストバージョン管理機能を提供するパッケージである。各クライアントは、バージョン管理の下にある固有のワークスペースとサーバ上に持つ。サーバは、この状態を全て管理することで、変更の伝播をはじめとするサービスを提供することになる。

(c) Basic-client–workspace Package

Basic-client–workspace Package は、バージョン管理下のリソースに関するデータをクライアントで取り扱うためのパッケージである。複数のクライアントが同一のリソースを並行して更

新する場合の一貫性を保つための機能として、元のリソースとは異なるワーキングリソースをサーバ上に配置する機能を提供する。このパッケージは、各クライアントが自分自身の名前空間を管理することで、サーバ側の実装を簡潔にする目的も果たしている。

2.5.5. バージョン管理機能の実現

WebDAV のバージョン管理機能は、WebDAV プロトコルの拡張として定義されさている。WebDAV の操作が XML 形式で記述されているのと同様に、様々なバージョン管理のための操作は、WebDAV 同様の XML 形式で送られ処理されることになる。

2.5.6. セキュリティ

ここでは、WebDAV のバージョン管理機能にかかわるセキュリティ上の検討事項について述べる。

バージョン管理機能は WebDAV および HTTP/1.1 を元にしているため、RFC 2518 で規定されている WebDAV の安全性に関する重要な問題は、バージョン管理についても当てはまる。

WebDAV では、バージョンヒストリーを管理することで、いつ誰が変更を行ったかを記録する。この機能により、不正なアクセスを検出する監査の仕組みを提供することができる。（ただし不正アクセス自体を防ぐことは別の問題である）

また、ユーザの間違いや不正な更新を防ぐために、WebDAV のバージョン管理機能では、バージョン管理下にあるリソースをあるバージョンで固定するという機能も提供している。この機能により、あるバージョンより前のバージョンの情報の保持を保証することが可能となり、間違いや不正更新からの回復および訂正を容易に行うことができる。

バージョン管理では、リソースと同様にバージョンヒストリーの管理にも注意を払わなければならない。バージョンヒストリーをリモートからアクセス可能にすることは非常に危険である。

上記の様な問題を含んでいるため、バージョン管理においては、より細かなアクセス制御が要求される。また、WebDAV のバージョン管理機能が提供する自動バージョン登録機能（オートバージョンニング機能）を悪用したサービス妨害攻撃に対しても注意を払う必要がある。自動的な新規バージョンの登録機能を利用すると、サーバ上に生成されるリソースが大きくなりファイ

ル空間を圧迫するため、サービス妨害攻撃の危険性をはらんでいる。また、クライアントの自動保存機能とこのサービスが結合することにより、予期せぬ事態が発生する危険性を含んでいる。

　このような問題があるため、ロックをかけたクライアントに対して、自動的な新規バージョン登録機能の利用を制限する等、不必要なバージョンの作成を減らす運用を行う必要がある。

2.6. ユーザとグループの認証

WebDAV は HTTP/1.1 の拡張プロトコルであるため、認証には HTTP/1.1 のメカニズムを利用している。HTTP/1.1 の認証メカニズムは RFC2617 において定義されている[16]。ここではその詳細について述べる。

2.6.1. 認証メカニズムの概要

HTTP/1.1 における認証には、Basic 認証と Digest 認証の 2 種類が存在する。以下これらについて述べる

(a) Basic 認証

Basic 認証は、その名前が示すように HTTP における最も基本的な認証方法であり、HTTP/1.0 から存在する。一般的な Web クライアント（ブラウザ）を用いて Basic 認証を要求する URL へアクセスした際の動作は次のようなものである。

（1） クライアントが目的 URL にアクセスする。このときクライアントは目的 URL において認証が必要であることを知り得ないためリクエストに Authorization ヘッダは付加しない。
（2） サーバは 401 Authorization Required をレスポンスする。なおこの時 WWW-Authenticate ヘッダとして認証タイプと、認証の領域（realm）が返される。
（3） クライアントはユーザに入力を促すなどして認証情報を用意し、Authorization ヘッダを付加したリクエストを再度要求する。
（4） サーバは認証処理を行い、許可された場合はコンテンツを送信する。

このように、Basic 認証ではリクエストヘッダに Authorization ヘッダを含めることにより処理が行われる。Authorization ヘッダは、ユーザ ID とパスワードをコロン（“：”）で連結し、それを Base64 エンコーディングしたものである。例を以下に示す。

---

[16] RFC2617 "HTTP Authentication: Basic and Digest Access Authentication" http://www.ietf.org/rfc/rfc2617.txt

```
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

(b) Digest 認証

Basic 認証の問題点は、パスワードが実際にネットワーク上で送信されることである。これはセキュリティ上非常に問題がある。これに対して Digest 認証では、MD5 方式によるチャレンジアンドレスポンス認証を行う。

MD5 はハッシュ関数であり、元文字列から一定長のハッシュ文字列を出力する。ハッシュ関数は一方向性があり、ハッシュ文字列から元の文字列へ逆変換することは困難とされている（不可能ではない）。

チャレンジアンドレスポンス認証は次のようなものである。

（1） 認証を行う両者の間で暗号系と鍵を打ち合わせる
（2） サーバがランダムなデータをクライアントに返す
（3） クライアントは受け取ったデータを、鍵で暗号化し、サーバに返す
（4） サーバも同様に送信したランダムデータを鍵で暗号化する
（5） 双方が暗号化した結果が同一であれば鍵も同一であり、正当なクライアントと判断することができる

ダイジェスト認証は両者を組み合わせたものである。通信シーケンスは次のようになる。

（1） サーバはあらかじめハッシュされたパスワードを持っている
（2） クライアントは Authorization ヘッダ無しでアクセスする
（3） アクセスがあると、サーバはランダムな文字列 nonce を作成し、WWW-Authenticate ヘッダに付加してクライアントにレスポンスする
（4） クライアントはユーザにパスワードの入力を促し、パスワードをハッシュする
（5） nonce とハッシュパスワードを規定の方法で組み合わせてハッシュしレスポンスを作成する
（6） クライアントはレスポンスを Authorization ヘッダに付加して再リクエストを行う
（7） サーバは保持しているハッシュ化されたパスワードと nonce を規定の方法で組み合わせてハッシュする
（8） クライアントのレスポンスと比較し、同一であればコンテンツを送信する

このシーケンスではセッションを識別することが必要であるので、ランダムな文字列 opaque をセッション ID として用いる。

(c) HTTP におけるユーザ識別

HTTP で用いられる認証方式では、認証に用いられる要素は「ユーザ名」「パスワード」「領域（realm）」の 3 種類である。

このうち、領域はその認証を識別する名前であり、クライアントがユーザを選択するヒントになる情報である。クライアントが認証を要求されるリソースにアクセスするたびにユーザ名とパスワードの入力をユーザに要求することを防ぐため、ある領域に対して一度認証されたユーザ名とパスワードは同一の領域にアクセスする限り再利用される。

したがって、HTTP におけるユーザ識別はユーザ名のみによって行われることになる。HTTP 仕様にはユーザにグループの概念は無く、グループに対する認証の概念も無い。

(d) グループに対する認証

実際には、ユーザのうちいくつかをグループとしてまとめ、そのグループに対してアクセスを許可するような認証方式が必要とされる。いくつかのサーバにおいてはグループに対する認証の設定が可能となっている。これはサーバ側でグループとそのメンバーのリストを持ち、認証の際に送信されてきたユーザ名が対象となるグループに含まれているかを調べ、次にパスワードを確認することにより認証を実現している。本質的には HTTP としてグループへの認証を行っているのではなく、サーバ側で認証のプロセスを一段増やしているにすぎない。

2.6.2. 実装

(a) HTTP ユーザの実装方法

HTTP ユーザをどのように定義しているかは、サーバソフトウエアによって異なる。

Apache では、ユーザは OS とは別個に Apache が持つデータベースにより定義される。そのため対応するモジュールを利用することにより、ユーザデータベースを gdbm や各種のリレーショナルデータベース、LDAP などで自由に管理することが可能である。HTTP ユーザは Apache が持つ以上の権限を持つことはできないが、セキュリティ上のメリットは大きい。

Windows サーバにおける標準的な Web サーバであるマイクロソフト社の Internet Information Server（以下 IIS とよぶ）では、HTTP ユーザは OS のユーザとして定義されている。つまり Windows のユーザとして登録されたユーザが HTTP の認証にも使用される。そのため Windows 独自のネットワークサービスについては適合し易いが、HTTP ユーザに OS にログオンしたのと同等の権限を与えるため、セキュリティ上のリスクも存在している。

(b) 認証方法の実装状況

HTTP/1.1 に定義される認証メカニズムの実装状況を以下に示す。

■ クライアントにおける実装状況

Basic 認証は HTTP/1.0 から存在する認証方法であり、ほぼすべての HTTP クライアントで実装されていると言ってよい。しかし Digest 認証をサポートしているのは比較的最近のソフトウエアに限られる。クライアントの実装状況を以下に示す。

表 2-1　クライアントにおける実装状況

名称・バージョン	Basic 認証	Digest 認証
InternetExplorer 4	○	×
InternetExplorer 5,6	○	○
Netscape 4,5,6	○	×
Netscape 7	○	○
Opera 6,7	○	○
Mozilla 0.9.6	○	×
Mozilla 0.9.7 以降	○	○
Cadaver	○	○
Web フォルダ	○	○
OS X ファインダ（10.1.3 以降）	○	○

■ サーバにおける実装状況

認証メカニズムのサーバにおける実装状況を以下に示す。

表 2-2　サーバにおける実装状況

名称・バージョン	Basic 認証	Digest 認証
Apache 2.0	○	○
IIS 5.0	○	△（注１）
OS X	○	○

※ 注１： IIS 5.0 におけるダイジェスト認証には、クライアントが IE であり、サーバとクライアントが同一または認証されたドメインに所属している必要があるという制約がある。

3. WebDAV を利用可能なソフトウエア

この章では、WebDAV 仕様をサポートするソフトウエアについて述べる。

3.1. 調査の対象

調査の対象としたソフトウエアを以下に示す。

（1） クライアントソフトウエア

(ア) Microsoft Windows 環境

(イ) Mac OS X

(ウ) UNIX（cadaver、sitecopy）

(エ) Dreamweaver、Acrobat 5

（2） サーバソフトウエア

(ア) Apache 2.0

(イ) IIS 5.0

(ウ) Mac OS X

(エ) Zope

3.2. クライアントソフトウエア

3.2.1. Microsoft Windows

（1） 概要

マイクロソフト社の Windows 環境には、複数の WebDAV 対応クライアントが存在する。

Windows 2000 および Windows XP には、Web フォルダへのアクセス機能が標準搭載されており、WebDAV プロトコルがデフォルトでサポートされている。

マイクロソフト社の Web ブラウザである Internet Explorer 5.0 以降や、統合ビジネスソフトである Office 2000 以降にも Web フォルダ機能は提供されている。Windows 9x や Windows NT ではこれらのソフトウエアを導入することで WebDAV の利用が可能になる。

Web フォルダへのアクセスはエクスプローラー上で行われ、ローカルのファイルと同様の GUI によるアクセスが可能である。

- 開発元
  米マイクロソフト社
- 配布形式
  バイナリ実行形式（Windows 2000 以降では OS に標準搭載）

（2） サポート状況

- DAV クラス 1 に準拠（Web フォルダ経由のアクセスにはロックは使用されない）
- SSL をサポート
- Microsoft Exchange 2000 Server との組み合わせにより、独自のクエリ文法に基づく DASL 機能の利用が可能

（3） 特記事項

2003 年 3 月に、Windows 2000 の WebDAV 機能を構成するコアコンポーネント（ntdll.dll）にバッファオーバーフローの脆弱性が報告されている[17]。セキュリティ対策として、マイクロソフト社より提供される修正プログラムを必ず適用すること。

Windows 2000 以前のクライアント機能には多言語対応に関し問題があり、日本語のリソース名を適切に取り扱えない。この問題は Windows XP においては解決されている。

---

[17] http://www.microsoft.com/japan/technet/security/bulletin/MS03-007.asp

3.2.2. Mac OS X

（1） 概要

アップル社の OS X 環境では、Finder から「サーバに接続」を起動し、afp ではなく http で接続することで WebDAV を用いたアクセスが可能になる。接続後はフォルダのウインドウが開き、通常のフォルダと同様に操作することができる。

- 開発元
  米アップルコンピューター社
- 配布形態
  バイナリ（OS に付属）

（2） サポート状況

- DAV クラス 2 をサポート（WebDAV フォルダに置かれたファイルをアプリケーションから編集する際には、WebDAV フォルダにより自動的にロックがかけられる）
- SSL をサポートしていない

（3） 特記事項

OS X では、「.DS_Store」という名前の不可視フォルダを作成し、ファイルに関するメタデータを保存しており、アクセス制限を行う必要がある。

OS X のバージョンが 10.2 未満の場合については、WebDAV サーバが Apache2 の場合に書き込みアクセスが失敗する既知の問題がある[18]。

バージョン 10.2 未満の Finder には

- WebDAV フォルダにゴミ箱を作成する
- Sherlock がインデックスファイルを作成する

といった仕様があり、ドットファイル全体に対してアクセス制限を適用すると問題が発生することが知られている。

[18] Mac OS X: Difficulty Using WebDAV Server
http://docs.info.apple.com/article.html?artnum=107047

3.2.3. UNIX 上のクライアント

(a) cadaver

（1） 概要

cadaver は UNIX 上で動作するコマンドラインベースの WebDAV クライアントである。ls、put、get、mkcol などの UNIX ライクなコマンドにより WebDAV リソースへアクセスすることができる。

調査時点の最新バージョンは 0.21.0 である。

- 開発元
  WebDAV Resources における cadaver プロジェクト[19]
- 配布形態
  ソースコード（GPL ライセンス）
  RPM 形式と deb 形式のパッケージも作成されている。

（2） サポート状況

- DAV クラス 2 に準拠
- SSL をサポート可能（コンパイル時に指定し OpenSSL ライブラリをリンク）
- DASL および Delta-V 拡張仕様をサポート

（3） 特記事項

cadaver は、WebDAV Resources において開発されている汎用の WebDAV ライブラリ neon を用いて実装されている。neon は WebDAV 仕様に熟知したメンバーが開発に携わり、各サーバとの互換性の問題についてもドキュメント化が行われているなど、実装に関しては高い信頼性を持つと考えられている。

cadaver は、WebDAV ユーザコミュニティにおいて、日本語化パッチが公開されている。

[19] http://www.webdav.org/cadaver/

3.2.4. その他のクライアント

(a) Dreamweaver MX

（1） 概要

Dreamweaver MX はマクロメディア社の Web サイト開発・管理ソフトウエアである。Dreamweaver ではローカルに保持するコンテンツを編集・管理し、その上でリモートの Web サーバ上のコンテンツと同期させる開発形態がある。Web サーバ上のコンテンツのアクセスに WebDAV プロトコルを用いることが可能である。

- 開発元
  米マクロメディア社
- 配布形態
  バイナリ

（2） サポート状況

- DAV クラス 2 をサポート
- SSL をサポート

3.3. サーバソフトウエア

3.3.1. Apache 2.0

（1） 概要

Apache は Apache Software Foundation において開発されているオープンソースの Web サーバソフトウエアであり、Web サーバソフトウエアとしては現在インターネットで最大のシェアを持つ。

Apache における DAV の実装は mod_dav という名称でモジュールとして実装されている。mod_dav の開発は WebDAV Resources において行われている[20]。Apache バージョン 1 においてはサードパーティ製モジュールの扱いであったが、Apache バージョン 2（Apache 2.0）では配布物に標準的に含まれている。

- 開発元
  Apache Software Foundation
- 配布形態
  ソースコード（Apache ライセンス）
  The Apache HTTP Server Project より取得可能[21]
  各種 UNIX 用のバイナリパッケージ、Windows 用バイナリも配布されている。

（2） サポート状況

- mod_dav 単体では DAV クラス 2 に準拠
- SSL に対応可能（Apache の SSL モジュールとの組み合わせによる）

（3） 特記事項

Apache はオープンソースでモジュール API が公開されているため、インターネットコミュニティにおいて多用な拡張モジュールが実装されている。DASL、ACL、Delta-V のモジュールを組み合わせることでそれぞれの拡張仕様が利用可能になる。

現状の mod_dav ではエンコードに UTF-8 のみをサポートしているので、それ以外のコードを扱うクライアントとの組み合わせでは問題が発生する。

---

[20] WebDAV Resources　http://www.webdav.org/
[21] The Apache HTTP Server Project　http://httpd.apache.org/

3.3.2. Internet Information Server（IIS） 5.0

（1） 概要

Internet Information Server はマイクロソフト社の Windows のコンポーネントとして提供されている Web サーバソフトウエアである（以下 IIS と呼ぶ）。WebDAV は IIS 5.0 以降でサポートされている。調査時点では Windows 2000 シリーズにおいて IIS 5.0 が、Windows XP においては IIS 5.1 が提供されている。また近い将来リリースされる Windows Server 2003 では IIS 6.0 が提供される。以下、本調査では IIS 5.0 を例にあげて記述する。

IIS の特徴としては、マイクロソフト社の他のコンポーネントとの密接な関連をあげるｋとができる。認証についても Windows のネットワーク認証に対応する機能が提供されている。WebDAV 関連では Exchange Server を用いた DASL 機能等が提供されているが、これらの独自の特徴は互換性の問題を含んでいる。

- 開発元
  米マイクロソフト社
- 配布形態
  バイナリ（OS のコアコンポーネントに含まれる）

（2） サポート状況

- DAV クラス 2 に準拠
- SSL をサポート
- Microsoft Exchange 2000 Server との組み合わせにより、独自のクエリ文法に基づく DASL 機能を提供可能

（3） 特記事項

2003 年 3 月に、WebDAV 機能を構成するコアコンポーネント（NTDLL.DLL）にバッファオーバーフローの深刻な脆弱性が報告されている[22]。セキュリティ対策として、マイクロソフト社より提供される修正プログラムを必ず適用すること。特に、IIS 5.0 が動作するサーバが、特別に構成された WebDAV への要求を受け取った場合、任意のコードが実行されうる。リモートからの攻撃者により、サーバの停止、システム権限による不正な操作等を受ける可能性がある。

アクセス制限を行う上では、IIS 5.0 においては Windows のユーザを利用してユーザ管理を行う。このためユーザ管理についても慎重な検討を行う必要がある。

---

[22] http://www.microsoft.com/japan/technet/security/bulletin/MS03-007.asp

3.3.3. Mac OS X

（1） 概要

Mac OS X にはクライアント版のほかにサーバ版が存在する。クライアント版は Apache を独自にインストールすることにより WebDAV が利用可能になる。サーバ版は OS のコンポーネントとして Apache が用意されており、WebDAV モジュールも組み込まれている。ただし、この Apache のバージョンは 1.3 系統のものである。

OS X Server における Apache の設定は、サーバ設定プログラムである「ServerAdmin」により可能である。

- 開発元
  米アップルコンピューター社（Apache1.3 を使用）
- 配布形態
  バイナリ（OS のコンポーネントに含まれる）

（2） サポート状況

- DAV クラス 2 に準拠
- SSL をサポート

（3） 特記事項

使用しているソフトウエアは Apache であるが、HTTP ユーザが OS と統合される等の独自化が施されている。

3.3.4. その他

(a) Zope

（1） 概要

Zope はオープンソースの Web アプリケーション環境である。Web アプリケーションの実行環境であると同時に開発環境でもあり、実行と開発の双方を容易にする豊富な機能を持っており、Web 上でダイナミックな Web アプリケーションの作成から運用までを行える。Zope は主にオブジェクト指向スクリプト言語 Python を用いて作成されており、移植性の高さも特徴に挙げられる。

Zope は独自の Web サーバを内蔵しており、現在のバージョン 2.6.1 では標準で WebDAV に対応している。Web サーバとして Apache とも組み合わせ可能であり、その場合の WebDAV 機能は Apache から提供されるため、Apache に準ずる。

- 開発元
  Zope Corporation および Zope コミュニティ
- 配布形態
  ソースコード（GPL ライセンス）
  各 OS にバイナリパッケージも配布されている。

（2） サポート状況

- DAV クラス 2 に準拠
- SSL をサポート

（3） 特記事項

Zope のコンテンツは動的に作成されるものが多いが、その場合はコンテンツのソースを修正する必要がある。もし Web オーサリングツールから WebDAV 経由でアクセスする場合は、sourcelink に対応している必要がある。アイコン画像など、静的なコンテンツの更新を WebDAV 経由で行う場合にはこのような問題は無い。

4. 類似するオーサリングツール

この章では WebDAV のバージョン管理機能に類似するオーサリングツールとして CVS を取り上げ、機能面に関して比較を行い、その整合性を示す。

4.1. CVS

ここでは CVS の概要について述べる。

4.1.1. CVS の概要

CVS は Concurrent Versions System の略であり、リモートアクセスにも対応可能なオープンソースのバージョン管理システムである[23]。バージョン管理システムは、プログラムのソースコードやテキストなどの各種のファイルに対し、バージョンの概念を適用し、特定のバージョンの取り出し、バージョン間の差分の取得などの機能を実現するものである。

CVS 以前の UNIX 環境では長く RCS（Revision Control System）が用いられていた。RCS は対象となるファイルのバージョン管理を行う機能を持っているが、ディレクトリごとに履歴情報を管理している、リモートアクセスができない、複数人による同時編集ができないなど、大規模なソフトウエア開発には問題があった。

CVS はバックエンドシステムとして RCS を用いている。CVS では rsh や ssh を用いたリモートアクセスへの対応、マージ機能による同時編集への対応などの拡張が行われた。これらの機能は分散環境における大規模ソフトウエア開発に適しており、特に開発者がインターネット上に分散して作業を行っているオープンソースソフトウエアの開発形態に適合している。そのため CVS は現在ではほとんどの UNIX 系オープンソースソフトウエア開発プロジェクトでソースコ

[23] http://www.cvshome.org/

ード管理に使用されており、Linux カーネルの開発、NetBSD や FreeBSD といった BSD 系 UNIX の開発においても開発基盤として用いられている。

また、レポジトリとローカルのファイルが分離されているという形態から、CVS を用いたリモートの Web サーバ上のコンテンツの編集も広く行われている。Web サーバ上のコンテンツを直接編集すると編集途中のコンテンツが公開されてしまうが、CVS を用いればローカルで編集を行い確認まで行った後でコミットすることでそのような問題を回避できる。

4.1.2. CVS の特徴

CVS の特徴は、リモートのレポジトリからチェックアウトして手元（ローカル）に置いたファイルを編集する点にあり、これにより編集時の更新の衝突が回避されている。さらに、編集を終えてレポジトリにコミットする際に、同時に行われていた他のユーザの編集との差分を取ってマージを行っており、これによりコンテンツの一貫性も保たれている。

また、CVS においては、RCS 由来のファイルごとのバージョン付けだけではなく、タグとブランチという概念が追加されている。

タグは、ある時点の各ファイルのバージョンに ID をつけ、タグの指定により複数ファイルをまとめて扱うことができるものである。これはソフトウエア開発において、開発途中のソースがうまく動作している時点のスナップショットをとり、チェックを行うなどの用途に適している。

ブランチは、タグの一種であるが、メインのバージョンの集合に対して分流となる集合を作成する機能である。ソフトウエア開発にブランチを用いることで、開発がひと段落するたびにブランチを分離させ、開発の本流となるブランチ（トランクと呼ばれる）とバグフィックスなどのメンテナンスのみを行うブランチを分けて管理することが可能になる。FreeBSD の開発ではこの機能を用いて開発ブランチと安定版ブランチを分け、開発ブランチで開発した新機能を安定後に安定版ブランチに取り込むという開発スタイルが採用されている。

4.1.3. WebDAV との比較

ここでは、WebDAV と CVS をバージョン管理機能のサポートおよび同一性（原始性）の観点から比較する。以下に比較表を示す。

表 4-1　WebDAV と CVS の機能比較

機能	WebDAV	CVS	同一性
チェックイン	○	○	○
チェックアウト	○	○	○
マージ	○	○	○
フォーク	○	○	○
前のバージョンの取り出し	○	○	○

主要な機能を取り上げる限り、WebDAV と CVS が提供している機能は同等であると考えられる。先に述べたように、CVS は幅広く利用されているバージョン管理システムであるため、WebDAV のバージョン管理機能も CVS の概念に適合するように設計されている。

4.2. CVS と WebDAV

CVS は現状ではバックエンドに RCS を用いている。しかし前項で検討したように WebDAV のバージョン管理機能と CVS は同等の機能を持っているので、リモートレポジトリを WebDAV で構築し、WebDAV のバージョン管理機能を有効にすることでバックエンドシステムに WebDAV を用いた CVS を実現することも可能である。2003 年 2 月時点においては、CVS のコミュニティにおいてその実現が検討されている。

参考文献

[1].The Internet Engineering Task Force

http://www.ietf.org/

[2].IETF WEBDAV Working Group

http://www.ietf.org/html.charters/webdav-charter.html

http://www.ics.uci.edu/~ejw/authoring/

http://ftp.ics.uci.edu/pub/ietf/webdav/

[3].WebDAV Resources

http://www.webdav.org/

[4].DASL Working Group

http://www.webdav.org/dasl/

[5].ACL Working Group

http://www.webdav.org/acl/

[6].Delta-V Working Group

http://www.webdav.org/deltav/

[7].World Wide Web Consortium

http://www.w3.org/

[8].Internationalized Resource Identifiers (IRIs) (2003/8/31)

http://www.ietf.org/internet-drafts/draft-duerst-iri-03.txt

[9].MSDN ライブラリ 「WebDAV の発行」

http://www.microsoft.com/japan/developer/library/default.asp?URL=/japan/developer/library/jpiis/core/wcwbdav.htm

[10].アップル - ソリューション - Web パブリッシング - WebDAV の設定 「Mac OS X Server の WebDAV を設定」
http://www.apple.co.jp/solutions/webpublishing/technology/webdavsetup/

[11].Zope Community
http://www.zope.org/

[12].Concurrent Versions System
http://www.cvshome.org/

[13].WebDAV Resources JP
http://webdav.todo.gr.jp/