分散オブジェクトによる抽象ドキュメントの管理方法*

2S－6

松島 英子†　瀬尾 明志‡
日本ユニシス株式会社§

1　はじめに

現在計算機上で扱われる文書には、HTML[1]をはじめ、LaTeX、PostScript、roffなど、様々な文書形式が存在している。これらはそれぞれに特徴があり有用なものである。しかしその半面、文書提供者にとっては多くの形式で文書を提供しなくてはならないという状況を発生させている。

このため文書形式の変換を行うツールは多数存在するが、文書形式毎に適当な変換ツールを選択し適応することは面倒な処理である。さらにその結果として、本来一つであるはずの文書に対して形式が異なる同一内容の文書が複数存在することになり、一元的な管理の面からも好ましくない。

また近年では、ネットワーク上での文書共有の要求が高まってきている。

そこで本稿では、必要とする形式の文書を取得するメソッドのみが公開されている「抽象ドキュメント」というオブジェクトを提案する。さらに、ネットワーク上での文書共有を可能にするために、抽象ドキュメントを分散オブジェクトとして実装する方法について述べる。

2　抽象ドキュメント

抽象ドキュメントとは、内部形式で書かれた文書と、その形式からHTMLなど他の文書形式への変換手続きをカプセル化したオブジェクトである。内部形式には、HTMLに幾つかのタグを追加し拡張したものを使用する。

内部形式文書を属性に、文書形式変換手続きをメソッドとして持たせることにする。文書作成者は、内部形式で文書を記述し、この文書を元に抽象ドキュメントを生成する。一旦抽象ドキュメントが生成されれば、それぞれの文書形式に応じた文書取得メソッドを呼び出すことにより、いつでも必要な形式の文書が取得できる。

また、文書を変更する場合には、内部形式文書の取得メソッドを用いて内部形式文書を取り出し、変更を加えた後に内部形式文書の更新メソッドを呼び出せば良い。

以下に抽象ドキュメントの属性とメソッドを記す。

- 属性

document　内部形式で書かれた文書。

password　文書保護のためのパスワード。オブジェクト生成時に登録する。内部形式文書の変更時に必要となる。

- メソッド

Constructor　document、passwordからオブジェクトを生成する。

changeDocument　新しいdocumentを用いて内部形式文書を更新する。このとき、passwordを検査することにより、不正な変更を防ぐ。

getDocument　内部形式文書を取得する。

以下は、文書取得のメソッドである。
実際の文書形式の変換処理自体は、できるだけ現在存在するツールを活用する。取得したい文書形式が増えた場合には、それに応じたメソッドを追加すればよい。

getPlaneText　プレーンテキスト文書の取得
getHTML　HTML形式文書の取得
getPostScript　PostScript形式文書の取得
getNroff　nroff形式文書の取得
getLaTeX　LaTeX形式文書の取得

このように、一つの文書に対して一つの抽象ドキュメントを対応させることにより、文書の一元管理が可能になる。文書内容を変更する場合も、作成者は内部形式文書のみを変更すればよく、今までのようにそれぞれの形式の文書まで更新する手間が省ける。また、文書の利用者は文書取得メソッドを呼び出すことに

*A management method of the abstract document with distributed object
†Eiko Matsushima
‡Akishi Seo
§Nihon Unisys Ltd.

より、常に最新の内容の文書を取得できることが保証される。

3 ネットワーク上における文書共有

ネットワーク上での文書共有を可能にするために、抽象ドキュメントを分散オブジェクトとして特定のマシン上に配置させる。このマシンを文書サーバマシンと呼び、ネットワーク上に複数存在してもよい。これに対して、文書の提供者や利用者が使用するマシンをユーザマシンと呼ぶことにする。

文書の提供者は、ユーザマシン上で作成した文書を元に、抽象ドキュメントを文書サーバマシン上に生成する。文書の利用者は、ユーザマシンから文書サーバマシン上にある抽象ドキュメントのメソッドを呼び出すことにより、必要な形式の文書を取得する。

また、HTML 形式文書の取得メソッドを CGI から呼び出すことにより、Web との連係も可能である。

4 分散オブジェクトとしての実装

抽象ドキュメントは Java(JDK1.1)[1] のオブジェクトとして実装し、分散オブジェクトを実現するための環境としては HORB(HORB 1.3)[2] [2] を用いることにした。

リモートマシンからのメソッド呼び出しは HORB を使用することにより可能になるが、抽象ドキュメントはさらに永続的なオブジェクトとして存在しなくてはならない。HORB における永続オブジェクトとは、オブジェクトの内容が保存されたファイルであり、オブジェクトを明示的にセーブ、ロードする必要がある。また、オブジェクトはメソッドの呼び出し側 (この場合ではユーザマシン側) に生成する方が自然なコーディングができる。

そこで、文書サーバマシンにはオブジェクトのセーブ、ロードを行うオブジェクトストレージという HORB の常駐オブジェクトを配置する。抽象ドキュメントは、常にユーザマシンに生成されるが、セーブ、ロードは常にこの常駐オブジェクトに依頼し、永続オブジェクトは文書サーバマシンに生成されるようにする。オブジェクトストレージは、永続オブジェクトにロックを掛けることができるので、一つの永続オブジェクトに対して同時に複数のオブジェクトがロードされることを防ぐことができる。

以上のことから、ユーザマシンでは、必要なメソッド呼び出しの前後に永続オブジェクトのセーブ、ロード、ロック等をオブジェクトストレージに依頼するという処理が必要となる。そこで、抽象ドキュメントの生成や、文書の取得など目的に応じた一連の処理をまとめたコマンドを用意した。文書の作成者や利用者はこれらのコマンドを使用することにより、必要な処理を簡単に行うことができる。

なお、現段階では文書取得メソッドの一部は完成していない。

5 まとめ

今回は内部形式として HTML を拡張した形式を採用した。しかし今後は、「内部形式文書は特定の内部形式で記述されたもの」という制限をなくし、「取得できる文書形式のいずれかで記述されているもの」とする予定である。その際には、「文書形式」という属性を追加することになるであろう。

さらに、現在は内部形式文書、文書取得メソッドによって取得される文書、ともに 1 ファイルとなっている。しかし一般の文書であれば、形式の変換とともにファイル数の増減が発生することになるであろう。この問題には、zip などのアーカイブツールを利用することにより対処する予定である。また、将来的には RCS、SCCS、CVS などを用いた文書のバージョン管理も行う必要があると考えている。

6 謝辞

本研究は情報処理振興事業協会 (IPA) が実施している「創造的ソフトウェア育成事業」の一環として行われたものである。

参考文献

[1] WORLD WIDE WEB Consosium, *Hyper-Text Markup Language (HTML)*, avairable at http://www.w3.org/pub/WWW/MarkUp/.

[2] HIRANO Satoshi, *HORB Flyer's Guid*,1996, available at http://ring.etl.go.jp/openlab/horb/.

[1] JDK は米国 Sun Microsystems,Inc. の製品です。

[2] HORB は通産省電子総合研究所の平野聡氏によって開発された Java のための分散オブジェクト技術です。