

2R-03 コンポーネントオブジェクト技術を利用したWebシステムにおける コンテキストデータストア機構の設計と実装

上原 智, 水野 修, 菊野 亨
大阪大学 大学院基礎工学研究科 情報数理系専攻
E-mail: s-uehara@ics.es.osaka-u.ac.jp

1 あらすじ

World Wide Web(以下Webと呼ぶ)システムの急速な普及に伴い, クライアントサーバシステムの開発プラットフォームとしてWebを採用するシステムが現在では主流となりつつある. 一方, Webシステムで用いられるHTTPプロトコルはセッションレスであるため, 複数のセッション間で共有すべきデータ(コンテキストデータ)を保持できないという問題がある. コンテキストデータを保持する為の機構は既に幾つか実現されているが, いずれも安全性や信頼性に関する問題点を抱えている. 我々は, コンポーネントオブジェクト技術を用いてそれらの問題点を解決する新たなコンテキストデータストア機構の設計と実装を行った.

2 研究の目的

2.1 Webシステムの問題点

一般にクライアントサーバシステムにおいては, 複数ページ間に依存関係があり, かつデータを共有するアプリケーションを構築することが多い. その場合, セッション間でのコンテキストデータを何らかの方法で保持する必要がある. 複数のセッション間でコンテキストデータの保持を目指す管理をセッション管理と呼ぶ.

Webシステムで用いられるHTTPプロトコルはセッションレスと言われ, 一回の通信(セッション)毎に通信が切断される. つまり, 図1に示すようにWebシステムでは時刻 t_1, t_2, t_3 での動作を連続したセッションとしてとらえることができない.

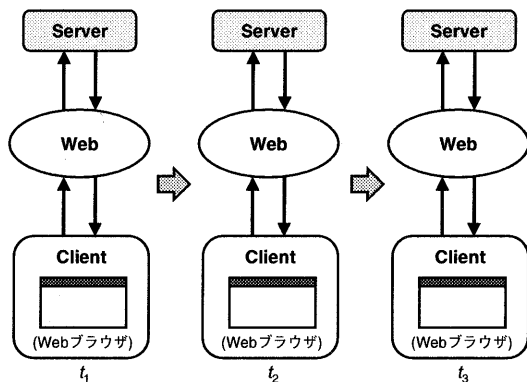


図 1: Webシステムの動作

また, Webブラウザでは, コンテンツを閲覧している最中にその内容が更新された場合, ユーザがWebブ

ブラウザの再読み込み機能を使って表示内容を更新しなければならない. そのため, 更新されるデータをWebブラウザ上にリアルタイムに表示することは非常に困難であった.

2.2 研究の目的

本研究の目的は, 2.1節で述べた問題点を解決するための新しいコンテキストデータストア(CDS)機構の開発にある. 3節で述べるように, セッション管理を行うための機構は既に幾つか提案されてきているが, 未だその特性は十分ではない. 提案するCDS機構では, 複数セッション間でのセッション管理を安全に, かつ, 高い信頼性を保ちながら実行できることを目指す. さらに, データの更新をCDSサーバから通知するだけで(ユーザからの更新作業を必要としない), 表示されるデータの更新が実行できるようにする.

3 現状技術とその問題点

Webシステム上でセッション管理を行う従来方式には, (1) 引数の引継ぎによるもの, (2) HTTPクッキー[1]を用いるものがある. ここで, HTTPクッキーと呼ばれる方式は, 通常, クライアント側にコンテキストデータを保存するものを指す. 一方, サーバ側にコンテキストデータを保存する方式もある. この2つを区別するために, 前者をクライアントサイドクッキー, 後者をサーバサイドクッキーと呼ぶ.

3.1 引数の引継ぎ

これは, HTML文書内にコンテキストデータを含める方式である. 特定のWebサーバやWebブラウザの機能に依存しないため, どのような環境でも利用できるという利点がある. しかし, 引継ぐべきコンテキストデータの内容がクライアント側のWebブラウザのアドレス欄やHTML文書内に暗号化されことなく現れてしまうため, セキュリティ上の問題がある.

3.2 クライアントサイドクッキー

これはクライアントコンピュータ側にクッキーファイルと呼ばれるテキストファイルを保存し, そこにコンテキストデータを書き込む方式である. この方式を利用するにはWebブラウザがその機能を備えている必要がある. コンテキストデータを提供する側で保存期間を指定できるため, Webブラウザを終了してもコンテキストデータは保存される. これを利用すれば, Webファームのアプリケーションを構築できる. しかし, クッキーファイルの内容は暗号化されないため, ユーザに保存場所が判明すると改ざんも可能である. したがって, やはりセキュリティ上の問題がある.

3.3 サーバサイドクッキー

これはコンテキストデータをWebサーバ側のメモリ上に保持する方式である。この方式では、クライアントコンピュータ側のクッキーファイルにはセッションIDしか保存されず、コンテキストデータの内容はクライアント側から見る事ができないため、セキュリティが高い。しかし、この方式では、サーバがダウンすると全てのコンテキストデータが失われてしまう。信頼性を上げる目的で複数のサーバによるロードバランシングを行うには、1つのWebブラウザは常に同じWebサーバにアクセスする機構が必要となる。それが実現できたとしても、やはりサーバダウンによるコンテキストデータの消失は避けられない。

4 提案技術

本節では、従来のコンテキストデータを保持する機構の問題点を解決する新しいコンテキストデータストア(以下、CDS)機構の提案を行う。

4.1 設計方針

提案するCDS機構は、前節までに述べた問題点を解決するために次の2つの特徴を持つように設計する。

- 1) クライアント側でのデータ保持：信頼性に関する理由から、提案するCDS機構ではコンテキストデータをクライアント側に保持する。
- 2) コンポーネントオブジェクト技術の利用：提案するCDS機構ではコンポーネントオブジェクト技術を利用して、コンテキストデータをクライアントコンピュータのメモリ上に保持する。また、Webブラウザ上で提案法を簡単に利用することができるようにする。更に、コンテキストデータの更新が行われた際にイベントを発行してWebブラウザに通知する機能を持たせる。

4.2 アーキテクチャ

提案するCDS機構はCDSサーバとCDSコンポーネントから構成される。CDSサーバはコンテキストデータを保持し、CDSコンポーネントはユーザインターフェイスとなる。

CDS機構を利用するには、HTML文書内にCDSコンポーネントを埋め込み、スクリプト言語を用いてCDSサーバに対するメソッド呼び出し、及び、イベントハンドリングを行う。CDSサーバはコンポーネントからの呼び出しに応じてコンテキストデータの保持や管理を行う。また、各コンポーネントにデータ更新の通知を行う。

図2にCDS機構の動作例を示す。時刻 t_1 では、CDSサーバがWebブラウザとは別プロセスで起動され、Webサーバから初期データ等を読み込む。時刻 t_2 ではコンテキストデータを参照するだけなので、Webサーバとの通信は行われず、CDSサーバだけがコンポーネントとコンテキストデータをやりとりする。ここで、CDSサーバはCDSコンポーネントに対してデータの更新を通知し、Webブラウザ上にリアルタイムに表示できる。時刻 t_3 でWebサーバへのデータ送信が行われ、一連のセッションが完了する。この時、Webブラウザ側でCDSコンポーネントを破棄するとCDSサーバも自動的に破棄される。

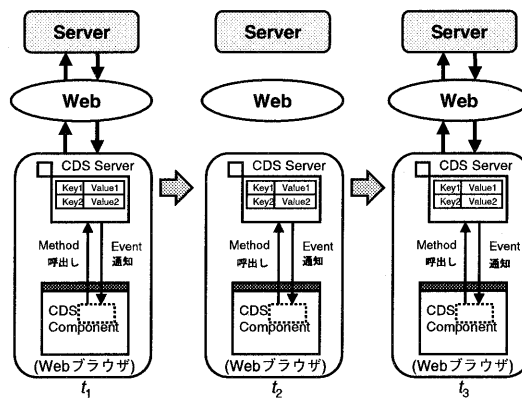


図 2: CDS 機構の動作

4.3 評価

ここでは提案するシステムの定性的な評価を行う。まず、4.1節の1)の設計方針で述べたように、利用するデータを全てクライアント側のCDSサーバで保持する。そのためコンテキストデータの一貫性を保つことができ、信頼性が向上する。

また、4.1節の2)で述べたように、コンポーネントオブジェクト技術を利用し、メモリ上にコンテキストデータを保持する。そのため、コンテキストデータの閲覧や改竄は困難になる。さらに、CDSサーバからクライアントへの通知機能により、Webブラウザ上でリアルタイムにデータ表示を行うことが可能になる。

4.4 実システムへの適用

前節で述べた設計に基づいてCDS機構を実装し、商用クライアントサーバシステムへの適用を行った。実装に際しては、Microsoftのコンポーネントオブジェクト技術であるCOM[2]を用いた。COM技術がMicrosoftの環境に特化した仕様となっているため、動作環境は、Microsoft Windows 95/98/MEまたはWindows NT 4.0/2000とMicrosoft Internet Explorer 4.0以上の組合せに限定している。

提案するCDS機構を適用した結果、従来の手法で同様の機能を実現した場合に比べ、はるかに少ない労力でシステムの構成が可能となった。また、経験的にはあるが、従来手法に比べ信頼性や安全性が強化されていることも確認できた。

5 まとめと今後の課題

本稿では、Webシステムにおけるコンテキストデータを保持するための新たな機構の開発を行った。新しい機構を利用すれば高い信頼性、安全性を持ったクライアントサーバシステムを容易に実現できると期待される。詳細は省いたが、実システムへの適用も既に行ってきている。今後の課題としては、提案したCDS機構の有効性の評価や、Microsoftに依存しない環境での実現が挙げられる。

参考文献

- [1] HTTP Cookies, http://home.netscape.com/newsref/std/cookie_spec.html
- [2] D. Rogerson, *Inside COM*, アスキー出版。