

Web-Based オペレーションへのコンポーネント技術の適応

5 J-2

藤田 元三郎 長島 雅夫  
NTT 情報通信研究所

1. 概要

現在、企業には多くのコンピュータが導入され、ネットワークを構成することで、企業活動の効率化が図られている。特に、WWWを用いたシステムは、イントラネット、インターネットの区別なく多くの企業、ユーザーに普及しており、オペレーションシステムの分野においても、近年、WWW関連技術を利用したシステム構成を採用するケースが増えている。

一方、システム開発環境としては、コンポーネントプログラミングを支援するRAD（Rapid Application Development）ツールを用いて、予め用意されたコンポーネントを組み合わせ、簡単なスクリプトを記述することによって、システムのカスタマイズが容易になりつつある。しかしながら、HTMLの記述力の限界によってWWW上の情報リソースを効率的にRADツールで再利用することが困難なことも明らかになってきた。

本稿では、Web-Basedオペレーションの現段階での問題点を分析すると共に、新たな発展型としてコンポーネント技術の適用を考察し、Web-Basedオペレーションの今後の可能性について言及する。

2. Web-Basedオペレーションの課題

オペレーションシステムへのWWW関連技術の採用は、要求仕様を容易に実現できるプラットフォームを開発者へ提供することを可能としている。その反面、異なるアプリケーションプラットフォームが混在して利用されるために、サーバーサイドでは複数システム間の連携を取る事が困難になってきている。この結果、カスタマに対してトータルなサービスオペレーションを提供するためには、それぞれのサーバーが提供する個々の機能を統合する必要性が生じている。一方、クライアントサイドにおいても、現在のブラウザを用いたまま、あるサーバーが提供する情報をもとに、他のサーバーへの情報検索・データ投入を行うといった機能連携を実現することは困難であり、その解決が求められている。

既に広く普及しているWWW関連リソースを有効に活用するためには、HTTPをベースにオペレーションシステム間の連携を行う方法についての検討が重要であり、その際、次の2点を考慮する必要がある。

- ・オペレーションシステム機能の連携は、ユーザーの要求仕様が多様化するために、ユーザーサイドでのカスタマイズが容易なアーキテクチャーを採用する必要がある。
- ・今後の分散オブジェクト環境の普及を考慮し、CORBA等を用いることでオブジェクトとして機能提供を行うオペレーションシステムとの連携も合わせて実現する。

3. オペレーション機能連携

本稿では、先述の条件を満足させるために、HTTPをベースとしたWeb-Basedオペレーションシステムの提供機能をコンポーネントプログラムを用いてラップすることにより、クライアントサイドで機能連携を行う手法を試みた。（図. 1参照）

クライアントサイドで機能連携を行うために、HTTPを用いて機能提供を行う個々のオペレーションシステムに対して各々コンポーネントオブジェクトを作成することとした。これらのコンポーネントはWebブラウザを用いた情報の受信およびオペレーションシステムへの操作を代行するProxyコンポーネントと位置づけられる。

一方、既にCORBAあるいはCOM等を利用して実装されたオペレーションシステムに対しては、Proxyコンポーネントを経由せずに直接RADツールに利用され、Proxyコンポーネントとの機能連携が行われる。

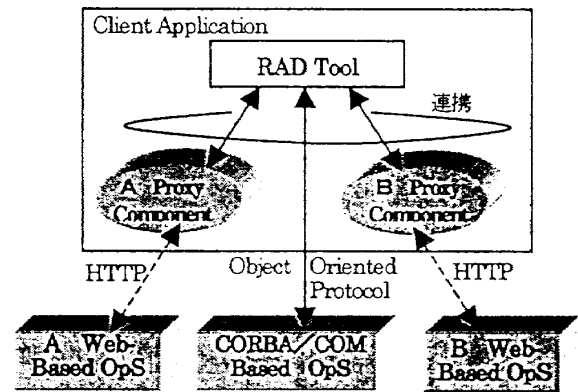


図. 1

4. Web-Basedオペレーションシステム

このようなProxyコンポーネントを実装するためには、HTTPを用いて送受信されるHTMLドキュメントの構文解析に加えて、ドキュメント中の各種タグの利用用途についての意味的解釈を行うことが必要である。しかし、HTMLではドキュメント中のデータ構造を明確に表わす手段が用意されていないために、意味的解釈はプログラマによるHTMLドキュメントのヒューリスティックな解析に大きく依存し、Proxyコンポーネントのコーディングを複雑にしている。さらに、WWWサーバー側でHTML記述を若干修正しただけで、Proxyコンポーネントの実装を変更する必要性が発生することも考えられる。

このようなHTMLの問題点を解決するために、現在、SGML（Standard Generalized Markup Language）をベースにその構文構造を簡略化し、WWWへの対応を考慮して定義されたドキュメント記述言語であるXML（eXtensible Markup Language）が注目されている。XMLの仕様はブラウザ以外の不特定の業務アプリケーションが必要とする情報を同ドキュメントから抽出可能にすることを目的に定義されているため、Proxyコンポーネントを

効率的かつ安定的に実装するために有効である。

以上を整理して、Web-Basedオペレーションシステムの構成を提案する。(図. 2参照)

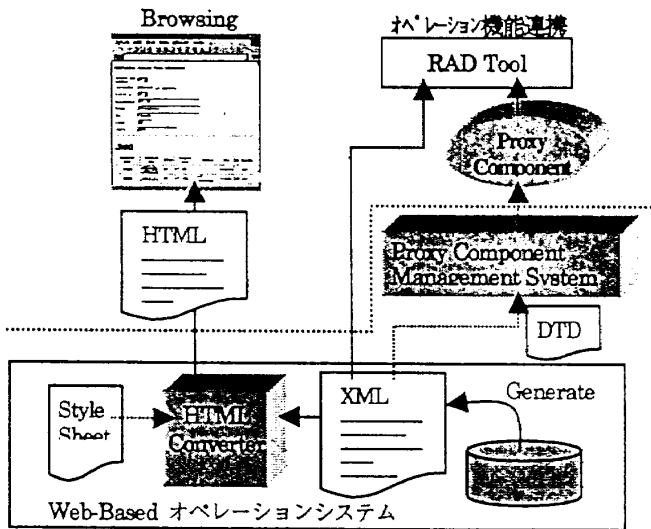


図. 2

Web-Basedオペレーションシステムは、データベース等の情報を利用して、RADツールを用いて実装されるクライアントアプリケーションあるいはブラウザからの要求に基づいたXMLドキュメントを生成する。クライアントがオペレーション機能連携を必要とする場合は、サーバーから直接XMLドキュメントを受信し、Proxyコンポーネントによって機能連携を実現する。また、クライアントサイドで既存のブラウザを用いてオペレーションを行う場合は、HTMLコンバーターがスタイルシートを参照することで、XMLドキュメントをHTMLドキュメントに変換した後、変換結果をクライアントに提供する。将来的には、XMLに対応したWebブラウザの普及が期待されており、その際にはHTMLコンバーターの必要性はなくなる。

なお、多数のWeb-Basedオペレーションシステムが個々のProxyコンポーネントを用意した場合、コンポーネント相互のインタフェースをさらに統一化するために、その生成・管理を一括して行う「Proxyコンポーネント管理システム」を外部に構築することが合理的である。

5. プロトタイプと評価

前項の構成に基づいてプロトタイプを構築し、その評価を行った。具体的なオペレーション対象としては、別々に構築されたカスタマアカウント管理システムとカスタマのホームページに関するアクセスログシステムをRADツールを用いて機能連携させる例を検討した。(図. 3参照)

本実験段階では、W3Cにおいてスタイルシートのオーソライズが行われていないため、スタイルシートが存在しないモデルとしてHTMLコンバーターを構築できるようにしたうえで、同コンバーターの独立性を高めるために一種のProxyサーバーとして実装を行うこととした。また、Proxyコンポーネントは、ActiveXコンポーネントとして実装し、RADツールとしてVisualBasic関連製品を利用できるものとした。なお、今回はス

モジュールモデルのためProxyコンポーネント管理システムの実装は対象外とした。

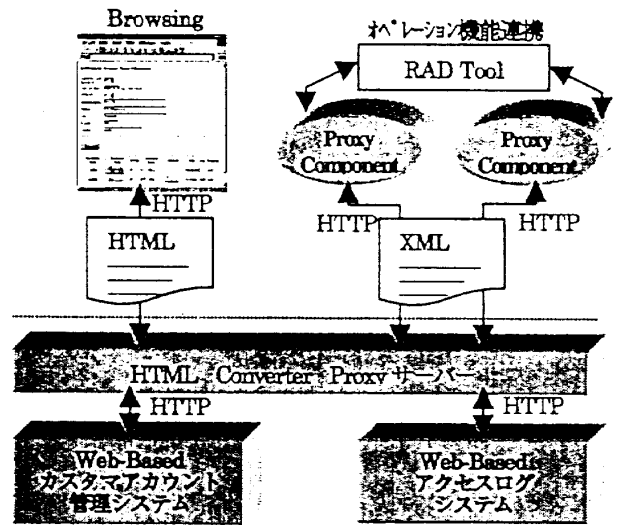


図. 3

プロトタイプを構築した結果、WWWサーバー側でXML記述に変更が生じた場合でも、Proxyコンポーネントプログラムの変更は生じず、Proxyコンポーネントの作成効率およびその安定性は、HTMLからの構築と比較して大きく向上することが判明した。

ただし、これはDTD(ドキュメントタイプ定義)が変更されない場合に限り、DTDが変更された場合はProxyコンポーネントプログラムも変更する必要があることが併せて明らかになった。この問題は、今回は実装されていないが、DTDを元にProxyコンポーネントの生成・変更などの管理を行う、Proxyコンポーネント管理システムによって解決する。

6. まとめ

以上より、現在のWeb-Basedオペレーションおよび本方式について評価・考察を行う。

- Web-Basedオペレーションシステムは、クライアントのユーザーインタフェースをWebブラウザに統一したという意味で、エンドユーザーに対してフレンドリーな環境を整えることに成功したが、一方で、現在普及しているブラウザでは、クライアントサイドでの複数サーバーとの機能連携を実現することは困難な状況にある。
- Web-Basedオペレーションシステムの提供機能をコンポーネントプログラムにラップする方法は、上記の問題を解決し、RADツールの利用を可能にするとともに、オブジェクト指向的に構築されたオペレーションシステムとの親和性を高める利点を持つ。
- 一方で、このようなコンポーネント(Proxyコンポーネント)を効率的に作成するためにはHTMLでは限界があり、XMLによるドキュメント流通への移行が必要である。
- 上記考察を踏まえ我々は、HTMLベースに若干の拡張を行ったXMLを使用するだけでも、コンポーネント作成効率およびその安定性が大きく向上することを明らかにした。