

企業情報ポータル基盤の研究 大規模組織対応機能の検討

菊地 克朗[†] 増岡 義政[†]

(株)日立製作所 中央研究所[†]

1. はじめに

企業内の情報システムは、従来のクライアントサーバ型システムから Web 技術を基盤とした Web アプリケーションに移行しつつある。これまでに様々な業務が Web アプリケーション化され、業務効率向上が図られてきている。その一方で、企業内には多数の Web アプリケーションが業務ごとに存在する結果となり、煩雑なログイン操作など業務システムの使い勝手が悪化している場合が多い。

上記問題の解決策として企業情報ポータル (Enterprise Information Portal、以降、EIP と略す) が注目されている。EIP では、業務システムをポートレットと呼ばれるラッパーで部品化し、ユーザが単一の画面に必要なに応じてポートレットを組み合わせられるようにする。これにより、ユーザは必要な情報に迅速にアクセスすることができるようになる。

2. 大規模組織における EIP 適用の課題

前章で述べたように、EIP 導入により必要な情報に迅速にアクセスすることができる環境が整いつつある。しかし、複数事業所から構成されていたり、グループ企業間で連携して事業を行っていたりするような大規模組織の場合、全ての Web アプリケーションを EIP により統合することは困難である。なぜなら、それぞれ独立した予算、拠点固有の業務システムを持つことが多いためである。そのため、事業所や関連会社毎に業務システムを統合したポータルサーバを構築、運用を行っているケースが多い。

たとえば、図 1 に示すように、本社、営業、開発部門があるような組織の場合、本社には総務系業務など全社共通の業務システムを統合する本社ポータルが、営業や設計部門にはそれぞれ固有の業務システムを統合するポータルが運

用される。そのため、ユーザは複数のポータルサービスの巡回を強いられる結果となり、そもそもの EIP の導入効果である「必要な情報に迅速にアクセス」が薄れてしまう傾向にある。

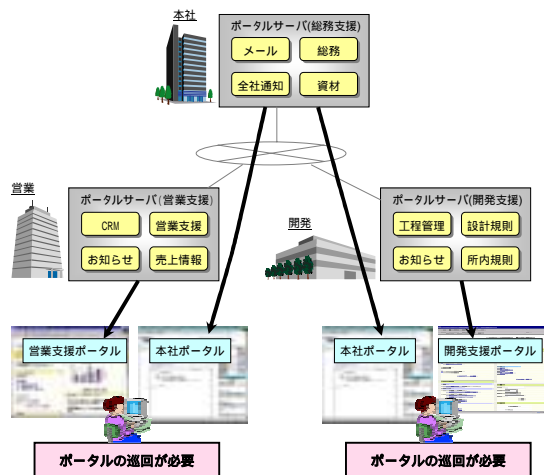


図 1 大規模組織における EIP 適用の課題

3. 解決案

全ての業務システムを集約し、単一のポータルとしてユーザに提供することができれば前記の問題は解決し、ユーザの利便性が向上するが、以下の理由から業務システムの集約は非現実的な場合が多い。

- ・ 業務システムは拠点ごとに運用管理
- ・ 大規模な計算機システムおよび広帯域ネットワークが必要

そこで、図 2 に示すようにポータルサーバ同士を連携させ、ポートレットをリモート呼出することにより、拠点ごとにポータルを運用しつつ、複数のポータルを統合し単一のポータルとしてユーザにサービスを提供することを可能とする。これにより散在する業務システムを集約することなく、全社統合ポータルの構築運用が可能となる。

リモートポートレットコールは、以下の 4 つの機能からなる。

(1) プロキシポートレット

ユーザのリクエストを RPC サーバ上で稼動するポートレットの実体(業務システム)へ転送す

る。呼出しには SOAP を利用する。

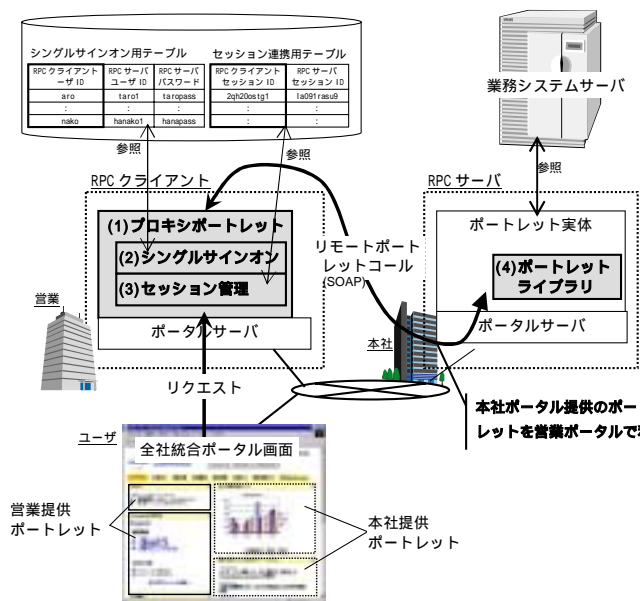


図 2 システム構成

(2) シングルサインオン

プロキシポートレットが RPC サーバへアクセスする際のログインを自動化し、ポータルサーバ間のシングルサインオンを実現する。

(3) セッション管理

ユーザ - RPC クライアント間のセッションと RPC クライアント - RPC サーバ間のセッションの関連付け管理を行う。

(4) ポートレットライブラリ

ポートレットのランタイムライブラリ。ポートレット開発者はリモートコールか、ローカルコールかを意識することなくライブラリを利用してポートレットを開発することができる。

4. 評価

今回採用した方式では、リモート呼出しが発生する分、ポートレット呼出しのオーバーヘッドが増加する。リモートコールの処理は、主にセッション管理、SOAP メソッド呼出しオーバーヘッドとなる。

セッション管理は、主にメモリ上のセッション連携用テーブルの検索となるのでマイクロ秒オーダーの処理時間でありオーバーヘッドとしては無視できる。一方、SOAP メソッド呼出しは XML データ解析等の処理が存在するので、ミリ秒単位のオーバーヘッドが発生すると考えられる。そこで SOAP メソッド呼出しのオーバーヘッドを測定し、リモートポートレットコールのオーバーヘッドを推定した。測定は、表 1 に示す環境を用い、

同一マシン内で何も処理を行わないメソッド呼出しを繰り返し実行することにより行った。

表 1 測定環境

CPU	Mobile Pentium III 800MHz
メモリ	512MB
OS	Windows 2000
JDK	Java 2 SE V1.3.1_02
SOAP エンジン	Apache AXIS beta 3

測定の結果、SOAP メソッド呼出しのオーバーヘッドは、20ms 程度であった。従って、リモートポートレットコールのオーバーヘッドは、20ms 程度であると推定できる。SOAP メソッド呼出しのオーバーヘッド測定結果より、ポータルサーバ間でポートレットをリモートで呼出すことによるオーバーヘッドの増加は、約 20ms 程度であると推定できた。本結果により、今回採用のリモートポートレットコール機能による、ユーザの使い勝手は悪化しない見通しを得た。

5. おわりに

本報告書で提案したリモートポートレットコール方式により、必要最小限の計算機およびネットワークリソースで、運用管理を複雑化することなく大規模組織対応のポータルサービスを実現できる見通しを得た。

参考文献

- [1] The Internet Engineering Task Force, "Hypertext Transfer Protocol -- HTTP/1.1", <http://www.ietf.org/rfc/rfc2616.txt>
- [2] World Wide Web Consortium, "Simple Object Access Protocol 1.1", <http://www.w3.org/TR/SOAP/>
- [3] Organization for the Advancement of Structured Information Standards, "Web Services for Remote Portals", <http://oasis-open.org/committees/wsrp/>
- [4] Organization for the Advancement of Structured Information Standards, "Web Services for Interactive Applications", <http://oasis-open.org/committees/wsia/>