

リモートポートレットの検索方式の研究 On the Search Engine for Remote Portlets

伊藤真澄[†]
Masumi Ito

奥野拓
Taku Okuno

大谷真
Makoto Oya

1. はじめに

現在、インターネット上には、ニュースや株価、天気予報などの各種コンテンツへアクセスするための集約型情報起点サイトとして多数のポータルサイトが開設されている。イントラネットの世界においても、ナレッジマネジメントを実現するためのツールとして企業情報ポータルを運営する企業が増えている。

今やポータルは、サービス提供者とエンドユーザとの間で中継役として各種サービス配信を行う代表的な情報ハブのデファクトスタンダードになりつつある。

この流れに伴って、2003年後半からポータル関連技術の標準化及びオープンソース開発が進行中である。

ポータル製品間で互換性が無かったポートレット/ポートレットコンテナ実装は、2003年10月のJSR168[1]標準化により、少なくともJava Platformにおいては可搬性が確保された。

インターネット上の分散したサーバに配置されるリモートポートレットに関する通信プロトコルの標準化とオープンソース実装も進んでいる。リモートポートレットの通信プロトコルは、2003年8月にWSRP[2]として標準化された。また、これに準拠したオープンソース・プロジェクトとして、Apache Software Foundationのインキュベータ・プロジェクトの一つであるWSRP4J[3]が2003年9月に開始され、現在開発の途中である。

一方、現行のWSRP及び開発途中のWSRP4Jではリモートポートレットの検索機能がなく、リモートポートレットの本格的な運用を考えた場合、問題である。

そこで本研究では、リモートポートレット登録/発見の仕組みに対する独自のアプローチとして、WSRPリモートポートレット検索エンジン Search Engine for Remote Portlets (SERP) のプロトタイプ開発、及びリモートポートレットサーバの多階層への拡張を検討した。

2. リモートポートレット

2.1 用語の定義

リモートポートレットとは、従来、ポータルと同じサーバ内に存在しなければならなかったポートレットを、インターネットを介して異なるサーバに配置できるようにする機能である。

また、実体を持たずに他のポートレットのプロキシとして働くポートレットを、プロキシポートレットと呼ぶ。本稿においては、以下の二つのサーバを定義する。

- リモートポートレットサーバ(RPS): リモートポートレットを配置するサーバ。
- ポータルサーバ(PS): 受け取ったリモートポートレットの生成する画面生成コードであるマークアップを集約し、ポータル画面を構成するサーバ。

なお、RPS - PS間には、WSRPプロトコルを用いて通信する。WSRPの定義に従うと、RPSがProducer、PSがConsumerに相当する。

図1にリモートポートレットの機能を利用したポータルのアーキテクチャを示す。

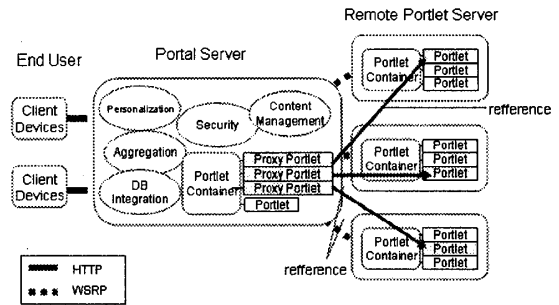


図1: リモートポートレットを利用したポータル

2.2 WSRP

WSRP(Web Services for Remote Portlets)は、リモートポートレットをWebサービスとして実現するためのプロトコルを規定したOASISの国際標準仕様である。

WSRPでは、Webサービス提供者のProducer、WebサービスクライアントのConsumer、Consumer利用者のEnd Userの三種類のアクタが定義され、Producer - Consumer間の通信プロトコルを規定するために、Producer側で統一された4つのインタフェースを備えることが義務付けられる。これらのインタフェースはWebサービスの記述言語であるWSDLによって記述される。

インタフェース	説明
Service Description	リモートポートレットのメタ情報の取得。(必須)
Markup	リモートポートレットのマークアップの取得及びユーザ・インタラクション処理。(必須)
Registration	ポータルとポートレットサーバ間の登録の確立・更新・削除。(任意)
Portlet Management	リモートポートレットのライフサイクルの管理。(任意)

図2: WSDLで定義された4つのインタフェース

2.3 解決すべき課題

現在、リモートポートレットを取り巻く問題点として、次の二点が挙げられる。

1. リモートポートレットの検索機能が無い。
2. WSRP準拠のポータルの普及に応じて、RPSやリモートポートレットの数が膨大になると、管理コストが増大する。

[†]北海道大学大学院情報科学研究科

問題点1について PS 管理者と RPS 管理者が異なるような、より一般的な状況においては、PS にリモートで接続する RPS やその RPS が保持するリモートポートレットの数が動的に時々刻々と変化していくため、リモートポートレットの検索機能がなければフレキシブルな運用が難しい。

以下に、検索機能の実現方法として考えられる二つの方法を示す。

- UDDI 等の Web サービスレジストリを用いた登録/発見のアプローチ
- 検索エンジンによるリモートポートレット探索のアプローチ

前者は、一般的に Web サービスで広く採用されているアプローチであり、標準化団体 OASIS の WSRP で検討が進められているが、後者については取り組みがなされていない。

そこで、後者の仕組みとして、独自のアプローチによる検索エンジン SERP を構築し、その概要を 3 章に示す。

問題点2について この解決策として提案するのが、RPS の多階層化である。多階層へ拡張することのメリットとして、リモートポートレットの追加/更新に関する一元管理の実現やセキュリティの強化などが挙げられる。一方、現状の WSRP は多階層を考慮した仕様になっていない。

そこで、多階層へ拡張するために必要とされる WSRP 仕様拡張の検討も同時に行った。これについては、4 章で述べる。

3. SERP の開発

3.1 機能

SERP とは、リモートポートレットのメタ情報を収集し、リモートポートレットの検索機能を提供する検索エンジンである。リモートポートレットのメタ情報には、

- ポートレット識別子 (ポートレット・ハンドル)
- RPS のエンドポイント URL
- ポートレットの提供する機能の説明やキーワード

等が含まれる。

3.2 処理の流れ

SERP の処理は基本的にロボット型 Web 検索エンジンと類似しており、図3で示されるように「収集」、「インデックス化」、「検索」の三段階フェーズ、それに「実行フェーズ」を加えた全体構造を持つ。

具体的には、以下の手順に従って動作する。

収集 WSRP プロトコルを利用して、SERP の検索ロボットである serpbots がリモートポートレットのメタ情報を収集する。

インデックス化 収集した情報を Indexer が検索用データとして加工する。この段階で、形態素解析、シソーラス作成、検索速度を速めるためのインデックス生成など、検索結果を高機能なものにするための複数の処理が行われる。

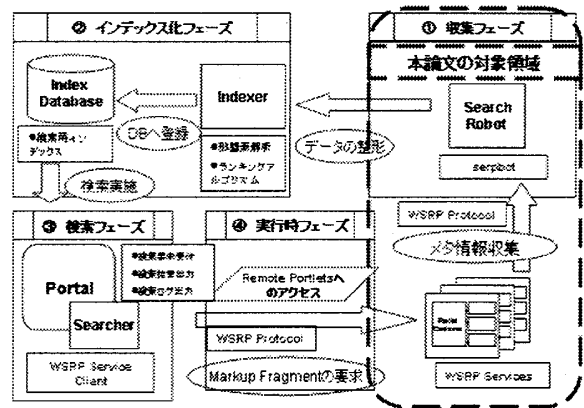


図3: SERPの全体構造

検索 検索用データの中から、Searcher が検索エンジン利用者の検索クエリに対応したデータを探し出し、検索結果として SERP の利用者にリモートポートレットのメタ情報の一覧を表示する。

実行時 Searcher によって返された検索結果を基に、検索エンジン利用者が、特定の RPS が保持する特定のリモートポートレットへマークアップを要求する。

3.3 収集フェーズのアルゴリズム

serpbots は、WSRP で定められたメタ情報取得の手順に準じた仕様である。次のような基本動作によって、インターネット上に分散したリモートポートレットのメタ情報を取得する。図4に、収集フェーズにおけるアルゴリズムを示す。

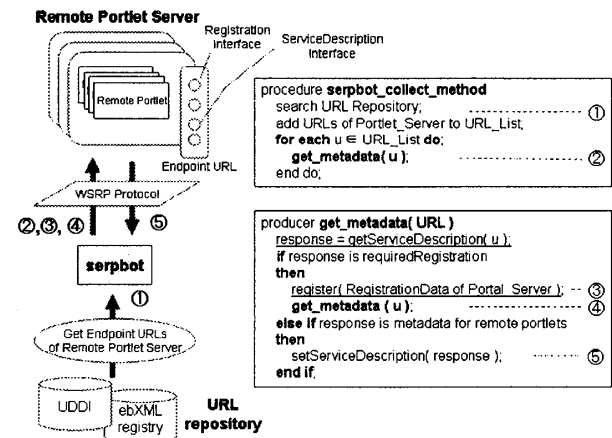


図4: 収集フェーズにおける serpbots の詳細

1. UDDI等のレジストリなどから、RPSのエンドポイント URLのみ取得する。(URL Repositoryとしての利用)
2. RPSのService Description インタフェースに対し、リモートポートレットのメタ情報を要求する。

3. RPSがRegistrationインタフェースを実装している場合、serpbot - RPS間の関係確立のため、「登録」作業を行う。
4. 「登録」作業終了後、再度メタ情報を要求する。
5. リモートポートレットのメタ情報を受け取る。

以上の動作を、繰り返し行う。

3.4 実装と評価

SERPのプロトタイプ実装には、WSRP4Jのプロダクトを利用した。具体的な実装方法としては、まず、WSRP4J Consumerのソースコードをベースにserpbotの開発を行った。ただし、エンドポイントURLの取得部分については、提案手法において本質ではないため、XML形式のURLリストをファイルより直接読み込む方法を用いた。また、インデクサ及びサーチャについては、あくまで簡易実装に留めた。

図5は、SERPの実行例のログである。

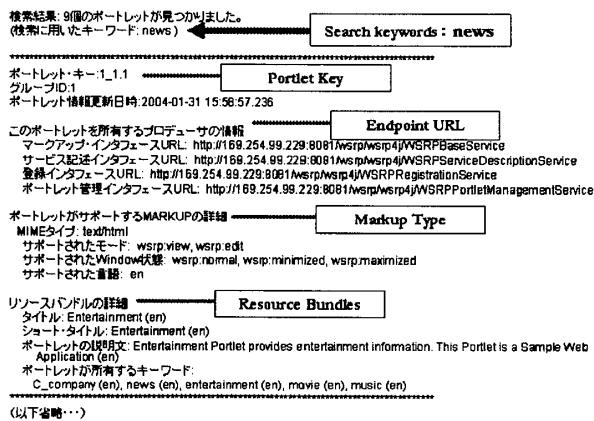


図5: 検索結果のログ情報

SERPを用いれば実時間内でリモートポートレットのメタ情報を収集できることが確認された。ただし、課題も幾つかある。

- Webページと異なりリンクの概念がないことから、既存のWeb検索エンジンのランキングアルゴリズムを用いることが出来ない。
- メタ情報とリモートポートレットの提供するコンテンツとの間で整合性の保障がなされていない。
- メタ情報作成の指針が無いため、異なるリモートポートレットに対して、同一内容のメタ情報が記述される可能性があり、メタ情報の重複が起こりうる。

これらは、SERPを実用に耐えうるものにするためには、解決しなければならない問題であり、今後検討が必要とされる。

4. 多階層への拡張

現行のWSRP仕様のままでは、RPSにリモートポートレットのメタ情報を要求すると、そのサーバが保持するポートレットがプロキシポートレットであった場合、

プロキシポートレットが指すポートレットの実体のメタ情報を受け取ることが出来ない。

そこで、この問題点を解決するために、メタ情報の取得方式として、以下の二つの方法を考案した。

透過型方式 (Transparent method) 再帰的にプロキシの参照先のポートレットまで、リモートポートレットのメタ情報取得要求を転送する方式。Consumerが受け取るメタ情報は、プロキシの参照するポートレットの実体のサービス記述である。

参照型方式 (Pointing method) プロキシの参照先のポインタを提示する方式。ポインタには、ポートレットハンドルとRPSのエンドポイントURLが含まれる。Consumerはプロキシポートレットから受け取ったポインタ情報を基に、次々にポインタを辿ってポートレットの実体を探索する。

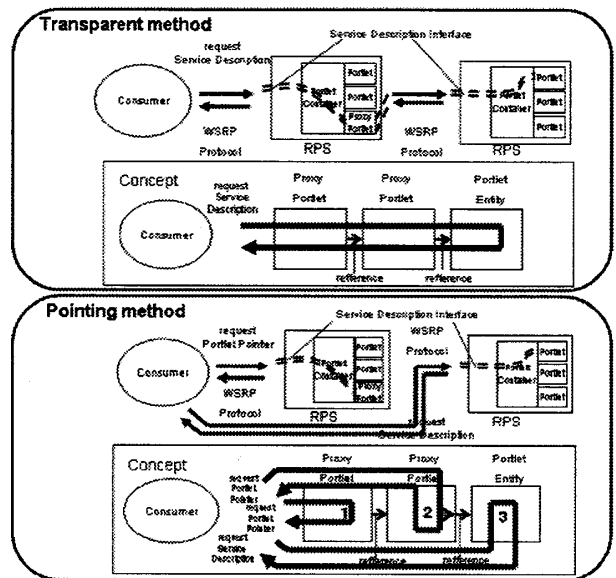


図6: 透過型方式と参照型方式

5. おわりに

以上より、SERPを用いたRPSにおけるリモートポートレットの検索方式の実現可能性について確認できた。今後は、ポータル関連技術の動向に注目しつつ、SERPの多階層RPSへの対応について取り組んでいきたい。

参考文献

- [1] Alejandro Abdelnur, Stefan Hepper, Java Portlet Specification version 1.0, Sun Microsystems, 2003
- [2] Alan Kropp, Carsten Leue, Rich Thompson, Committee Specification 1.0: OASIS Web Services for Remote Portlets, OASIS, 2003.
- [3] WSRP4J H.P., <http://ws.apache.org/wsrp4j/>