

ワークフロー管理システムを対象とした Web サービス化の試作

関 洋子 秋藤 俊介

(株)日立製作所 システム開発研究所

1. はじめに

近年、多くのベンダが Web サービスの開発・実行ツールを提供し始めている。これらのツールには、既存のアプリケーション・システムの Application Programming Interface(API) から、Web サービスのインタフェースを記述する Web Services Description Language(WSDL)や、通信プロトコル Simple Object Access Protocol(SOAP)で接続するためのプロキシ・プログラムを自動的に生成する機能がある。しかし、実用のアプリケーション・システムが Web サービス化された例は、まだ少ない。

本報告では、アプリケーション・システムの 1 つとしてワークフロー製品を対象とした Web サービス化の試作を行った。既存 API を Web サービス化する際の課題と、その解決方法を報告する。

2. ワークフロー管理システム API の概要

Web サービス化の対象として、日立 Groupmax Workflow^[1]の Active Server Pages(ASP)ライブラリ Workflow for ASP が提供する API (以下 ASP API とする) を対象とした。ASP API は、Web サーバ上で実行可能なコンポーネントであり、オブジェクト・インスタンスに対する操作を行うメソッドと、値を設定・取得するためのプロパティにより構成される。具体的には、案件投入、案件遷移、案件情報取得などワークフローの基本操作を行う基本メソッドと、各基本メソッドを実行するための準備を行うサブメソッドおよびプロパティを有する。

3. Web サービス化の課題

ASP API の内部に手を加えずに SOAP 接続を可能にするためのためのラッパーを開発することとした。

ASP API の各メソッド・プロパティと 1 対 1 に対応する新規 API (以下 SOAP API とする) をラッパーとして開発しようとする場合、以下のような課題がある。

(1) セッション保持の問題

ASP API は、接続中のオブジェクト状態を保持

するため、セッションフルの仕組みを有している。一方 SOAP は、クライアント/サーバ間でセッションを共有する仕組みが標準化されていないため、セッションの保持が困難である。

(2) プロパティの Web サービス化が困難

Web サービスの単位処理は、何らかの操作を行うオペレーションである。ASP API の API のうち、メソッドの形を取る API はそのまま、Web サービスのオペレーションに対応付けることができる。しかし、プロパティの形を取る API については、現状で存在する Web サービス開発・実行ツールでは、Web サービスとしてデプロイしたり実行したりすることはできない。

(3) SOAP 接続による性能の問題

SOAP 接続は XML 処理を伴うため、オブジェクトを直接操作する ASP API に比べ、処理速度の低下が懸念される。ASP API のように 1 つの処理を実行するのに複数回の接続を行うのは望ましくない。

4. 課題の解決方法

3. に述べた課題解決のために、ASP API と 1 対 1 に対応させるのではなく、ワークフロー接続からメソッド・プロパティ実行およびワークフロー切断までの一連の処理を 1 つのメソッドとしてラップするラッパーを開発した(図 1)。具体的には、SOAP API を、以下のように設計した。

(1) メソッドの引数

SOAP API のメソッドの引数は、ASP API の形式にできるだけ近い形にするため、ASP API の基本メソッドの引数をそのまま与えるものとした。それに加え、ASP API の基本メソッド実行の準備として設定していたサブメソッドの引数やプロパティの値も、SOAP API のメソッドの引数として追加した。

(2) メソッドが行う処理

SOAP API のメソッドは、ラップ対象となる ASP API の基本メソッドと同様の順序で、サブメソッド・プロパティおよび基本メソッドを順に呼び出す。

(3) メソッドの戻り値

ASP API には、実行後の結果を、プロパティにより取得できるような基本メソッドがある。取

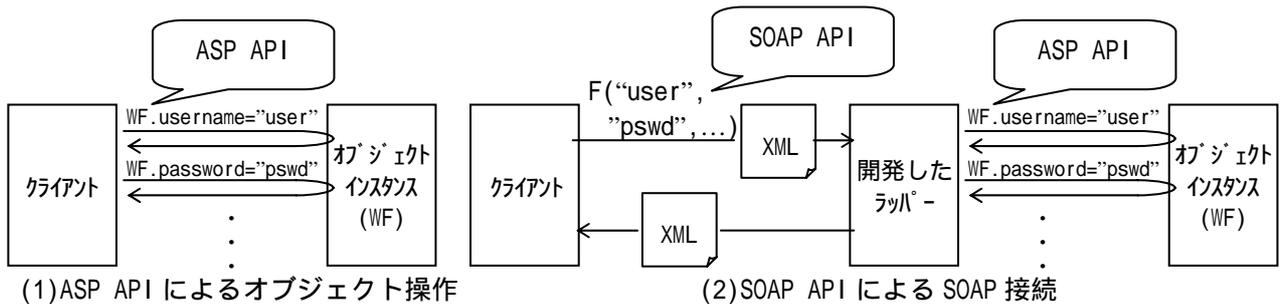


図 1. 接続方式の比較

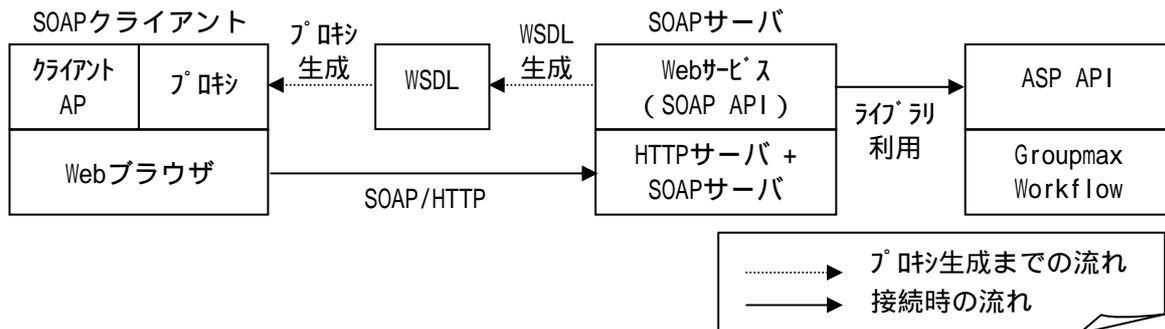


図 2. システム構成

得結果はメソッドの引数やプロパティ値から推測できるため、結果を XML ドキュメント形式で返す設計にし、そのテンプレートを定義した。

ワークフロー接続から目的の操作を経てワークフローを切断するまでの処理を、1 回の SOAP 接続で完了する。また、SOAP API の各メソッド内に、ASP API で行っていたプロパティによる値の設定・取得も隠蔽するため、プロパティを直接 Web サービスとしてデプロイしたり実行したりする必要はなくなる。

5. プロトタイプ開発

5.1 プロトタイプの概要

ASP API が提供する 58 の基本メソッドのうち、Web サービス化する 32 メソッドを選別した。本報告では、32 メソッドのうち案件投入、案件遷移、案件情報取得を含む 18 メソッドと、そのサブメソッド・プロパティとをラップする新規メソッドを、SOAP API として実装した。開発言語には C#を用いた。開発したラッパーを Web サービスとしてデプロイし、WSDL およびプロキシ・クラスを Web サービス・ツールにより自動的に取得した。プロキシおよびクライアント AP は、WSDL を解釈するなら原則的にはどのクライアント環境で開発しても良い。本報告では ASPX ファイルとして試作し、SOAP プロトコルによる接続を確認した(図 2)。

5.2 プロトタイプの性能評価

ASP API によるオブジェクト操作と、SOAP API による SOAP 接続の、要求応答時間を計測した。計測時間は、メソッドの実行直前から直後までの時間を計測し、プロキシ・インスタンス生成など、その他の処理に要する時間は含めない。

案件情報取得メソッドの処理を例に取り、ASP API と SOAP API の要求応答速度を比較した。ASP API では平均して 80msec 程度、SOAP API では 90msec 程度であった。SOAP 接続の方が 10msec 程度遅くはなるが、実用レベルでの利用に、大きな問題はないと考える。

6. おわりに

ワークフロー管理システムを例に取り、既存アプリケーション・システムの API を Web サービス化した。API ごとにサービスを対応すると、性能上の問題やセッション保持などの課題が生じる。本報告では、API の処理の意味と接続性を分析し、複数 API を 1 つの API としてラップするラッパーを、Web サービスとして開発した。本方式により、SOAP 接続の課題の 1 つである処理速度の低下も概ね解決することができた。

参考文献

- [1] 宍戸,他:Groupmax ワークフローによる業務革命の真髄, IDG コミュニケーションズ, (1999).