

## 組込み Web サービス構築環境 $\mu$ WS-Suite

### $\mu$ WS-Suite : Embedded Web Services Building Environment

阿部 一裕† 炭崎 竜平† 荒井 兼秀† 遠藤 義雄†  
Kazuhiro Abe Ryuhei Sumisaki Kanehide Arai Yoshio Endo

#### 1. はじめに

産業システムでは、生産管理・保守・エネルギー管理・CRM といった観点から、フィールド情報の収集、管理が行われてきた。インターネット/イントラネットの普及に伴い、これらの業務の高度化(より多く、より詳細に、より早く、どこからでも)のニーズが高まっており、インターネット/イントラネットを介し、オンデマンドに、フィールドネットワーク上の組込み機器の情報を取得し、機器の設定を行う機能が求められている。

このような背景のもと、我々は、フィールドネットワーク上の組込み機器に Web サービス機能を実現し、生産管理や受発注管理等の業務系システムとのシームレスな連携を図ることを目的として、組込み Web サービス構築環境  $\mu$ WS-Suite の開発を行っている。

組込み機器に Web サービス機能を実現する場合、限られたメモリ・CPU リソースで Web サービスの通信プロトコルを処理する SOAP メッセージ処理部を実現し、サービスを実体である組込み機器のアプリケーションソフトウェアを、SOAP メッセージ処理部から利用する機能を開発する必要がある。また組込み機器が提供する Web サービスのインタフェースを定義する WSDL ファイルを記述し公開する必要がある。組込み機器開発者が、Web サービス技術の詳細に通じていない場合、これらの開発は容易でなく開発コストの増大をもたらす。

本稿では、上記問題を解決するために、 $\mu$ WS-Suite の一環として開発した、組込み機器向け SOAP サーバである  $\mu$ SOAP サーバと、Web サービス生成ツール  $\mu$ WS-Gen の設計と実装について述べる。

#### 2. $\mu$ SOAP サーバ

$\mu$ SOAP サーバは、Web サービスの標準通信プロトコルである SOAP のうち組込み機器に必須の機能を省メモリで実装した組込み機器向け SOAP サーバである。

組込み機器の限られたメモリ・CPU リソースで動作可能で、組込み機器が動作する多様な実行環境での動作に対処するために、下記に示す方針のもと設計・実装を行った。

- a) SOAP 仕様のうち組込み機器向けに必須な機能を実現
  - SOAP の 4 つのメッセージパターンのうち Request/Response 型のメッセージパターンのみを取り扱う。
  - 入出力パラメータの型は、単純型として short, int, long, float, double, String を、複合型として short, int, long, float, double の一次元配列を対象とする。
- b) モジュール化されたソフトウェア構成
  - 図 2 に示すように HTTP モジュール、XML モジュール、SOAP モジュール及び、後述する  $\mu$ WS-Gen により自動生成される Glue コードの各部からなるモジュール化されたソフトウェア構成を取る。

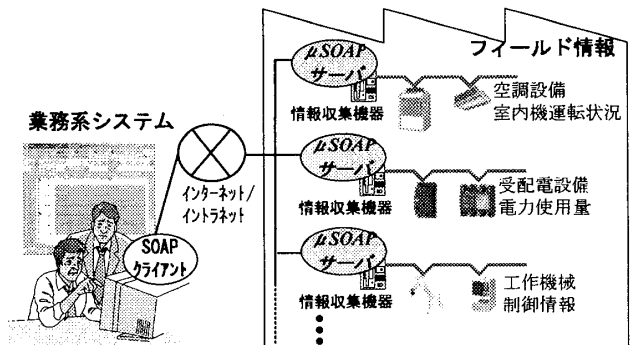


図1 フィールド機器と業務系システムの連携

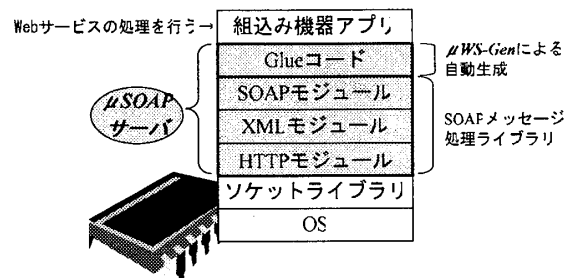


図2  $\mu$ SOAP サーバのソフトウェア構成

- 産業システムで用いられる機器では IP ネットワーク以外のフィールドネットワークに接続される場合もあるが、このような機器にポーティングする場合は、HTTP モジュール以下を、フィールドネットワーク向け通信モジュールに置き換えることにより対応可能である。

#### c) 省メモリ実装

- 組込み機器のソフトウェア開発で使用頻度が高い C 言語を用いて実装した。必須なライブラリは TCP/IP 通信を実現するソケットライブラリだけであり、多量のリソースを必要とする JavaVM, Servlet エンジン等ミドルウェアの組み込み機器への実装は不要である。
- XML 文書の解析のために、DOM と比べメモリ使用量が少ない SAX ベースのパースを実装した。このパースでは SOAP メッセージの解釈に必要な機能のみを実装している。

SOAP クライアントとなる業務系システムでは、Microsoft Visual C++, Visual Basic 或いは Java にて構築されるケースが多い。これら環境からの  $\mu$ SOAP サーバへのアクセスを想定して、下記の考慮も行った。

#### d) SOAP クライアントツールとの相互運用性の確保

- SOAP メッセージでの入出力パラメータの型表現を SOAP クライアント側スタブクラス生成ツールである Apache AXIS wsdl2java[1]、Microsoft Visual Studio .NET

† 三菱電機 (株) 先端技術総合研究所

[2]の両者が生成するスタブクラスで処理可能でかつ、簡潔な形式を取る表現を用いることとした。後述する  $\mu$ WS-Gen が出力する WSDL ファイルで、これらの形式でパラメータ型が表現されるよう定義しておくことにより、WSDL ファイルより生成される SOAP クライアント側スタブクラスでも、SOAP メッセージ中のパラメータ型に上記の表現を使用させることが可能となる。SOAP メッセージを伝送する HTTP に関しては、リクエストメッセージに対しては HTTP1.0 と HTTP1.1 のいずれにも、レスポンスメッセージに関しては HTTP 1.1 に対応することとした。

- 準拠する SOAP のバージョンは SOAP1.2[3]とした。

上記方針のもと、実装した  $\mu$ SOAP サーバの現状は、下記の通りである。

- CPU に Intel Pentium4、OS に Red Hat Linux Work Station を用いた環境下で、単純な算出演算を行う関数に対して、後述する  $\mu$ WS-Gen にて Glue コードを生成し、これらと  $\mu$ SOAP サーバライブラリを gcc -Os でコンパルしたところ、実行オブジェクトのサイズは約 24KByte、TEXT+DATA で約 10KByte、BSS が約 68KByte であり、スタックサイズ 20KByte に制限した環境下にて実行可能であることを確認した。
- 動作確認している CPU は、SH4、MIPS、x86 である。
- 動作実績のある OS は、ITRON、VxWorks、Linux、Windows 2000/XP である。
- 相互運用性を確認している SOAP クライアント側スタブクラス生成ツールは、Apache AXIS 1.2beta wsdl2java、Microsoft Visual Studio .NET2003、Microsoft Office SOAP 3.0[4]である。

### 3. $\mu$ WS-Gen

$\mu$ WS-Gen は、組込み機器開発者が、Web サービス技術の詳細に通じていない場合でも、容易に組み込み機器の Web サービス化を実現することを可能とする Web サービス生成ツールである。

$\mu$ WS-Gen の処理の流れを図3に示す。ユーザは図4に示す GUI を通し、①Web サービスの処理を行う組込み機器のアプリケーションソフトウェアのインタフェースを記述した C 言語のプロトタイプ宣言ファイルを指定する。②同ファイルでのプロトタイプ宣言のうち、Web サービスとして公開する関数を選択し、③SOAP メッセージ中で用いる サービス名、パラメータ名等を示すタグ名、URI、Name Space 等の通信パラメータを指定する。

$\mu$ WS-Gen は、これらの情報を用いて、Web サービスのインタフェースを記述した WSDL ファイルと、 $\mu$ SOAP サーバから組込み機器のアプリケーションソフトウェアモジュールを用いるための Glue コードを生成する。

組込み機器開発者は、 $\mu$ SOAP サーバライブラリ、Glue コード、組込み機器のアプリケーションソフトウェアモジュールをリンクして  $\mu$ SOAP サーバを構築し、組込み機器に搭載する。またクライアント側システムの開発者に WSDL ファイルを配布する。

クライアントシステム開発者は WSDL ファイルより、Apache AXIS wsdl2java、Visual Studio .NET 等のツールを用いて SOAP クライアント側スタブクラスを生成し  $\mu$ SOAP サーバが搭載された組込み機器にアクセスするアプリケーションを開発する。

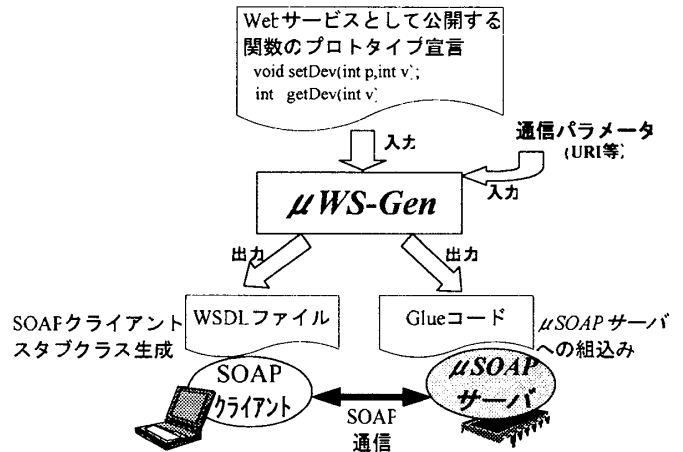


図3  $\mu$ WS-Gen の処理の流れ

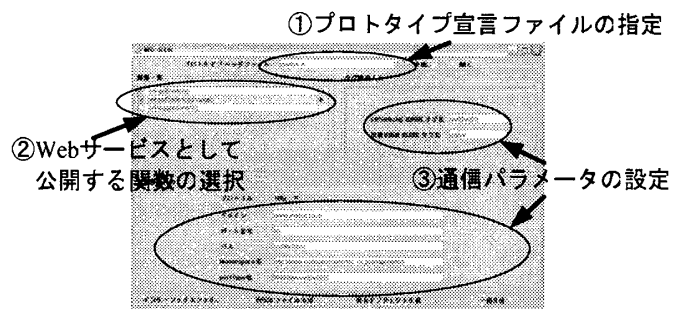


図4  $\mu$ WS-Gen の GUI

### 4. おわりに

本稿では、組込み Web サービス構築環境  $\mu$ WS-Suite の一環として開発した、 $\mu$ SOAP サーバと、 $\mu$ WS-Gen の設計と実装について述べた。

設計・実装上の工夫により、 $\mu$ SOAP サーバと同様に C で実装された gSOAP[5]と比較し、実行オブジェクトサイズで 1/3、実行時に必要なメモリサイズで 2/3 の容量に削減することができた。また WSDL ファイルと Glue コードの生成を支援する  $\mu$ WS-Gen を用い、組込み機器に Web サービス機能を実現する方式を開発できた。

今後、組込み機器間で公開された Web サービスに対するアクセス制御[6]を考慮した構成管理機能、IP ネットワークとは異なるフィールドネットワーク上の組込み機器に対する SOAP 通信機能等の開発を進めていく予定である。

### 【参考文献】

- [1] Apache Project: WebServices Axis, URL: <http://ws.apache.org/axis/> (2004).
- [2] Microsoft: Visual Studio, URL: <http://www.microsoft.com/japan/msdn/vstudio/default.aspx>(2004).
- [3] W3C: SOAP Version 1.2 Part 0: Primer, URL: <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/> (2003).
- [4] Microsoft: Microsoft Office 2003 Web Services Toolkit の新機能, URL: [http://www.microsoft.com/japan/msdn/office/dev/wst2003/dc\\_WSTKNew.asp](http://www.microsoft.com/japan/msdn/office/dev/wst2003/dc_WSTKNew.asp)(2004).
- [5] Engelen, R.: gSOAPFeatures, URL: <http://www.cs.fsu.edu/~engelen/soapfeatures.html>(2004).
- [6] 炭崎他: Web サービス機能を搭載した組込み機器のためのアクセス管理方式, FIT2004(2004).