

ユビキタスネットワークにおける 組み込み機器向け分散プロトコル処理

Distributed Protocol Processing for Simple Embedded Devices in a Ubiquitous Network Environment

荒井兼秀十 遠藤義雄十 炭崎竜平十
Kanehide Arai Yoshio Endo Ryuhei Sumisaki

1. はじめに

近年、情報家電やセンサなどの多数の組み込み機器がネットワーク接続され相互通信を行うユビキタスネットワーク環境の研究がなされている。このような環境を構成する組み込み機器は多数のベンダから提供され、アーキテクチャや適用される通信媒体も多岐に渡る。そのため、その通信プロトコルはオープンで通信媒体や機器アーキテクチャに非依存であることが求められる。一方、インターネットでの Web サービスを実現する通信プロトコルとして SOAP[1]がある。SOAP は、上記条件を満たす通信プロトコルであり、組み込み機器適用のための実装も行われている[2]。

しかし、通常、組み込み機器はそのメモリ容量の制限が厳しく、システムを構成する組み込み機器に XML パーサを含む SOAP 通信機能を搭載するのが困難な場合がある。このような場合、一般に、プロキシサーバを機器とクライアントとの間に配置し、プロトコル変換を行う方式がとられる[3]。この方式では、クライアントが、機器ではなくプロキシサーバにアクセスする必要がある。例えば、RS485 上で組み込み機器同士が通信する場合や、IPv6 環境でセキュアな通信をクライアントと機器とが通信する場合に、SOAP 通信を行うクライアントは、通信相手機器(サーバ)が SOAP 通信を行わないことを知り、プロキシサーバにアクセスする必要がある。

上記課題を解決するため、本稿では、XML パーシング処理は機器に特化した処理でないこと、機器に依存した情報は WSDL(Web Services Description Language)ファイルで定義されること、小規模な組み込み機器では、その扱うデータ型が単純であることが多いことに着目し、XML パーサ機能を組み込み機器外に配置する分散プロトコル処理方式を提案する。本方式により、クライアントに対して、XML パーサ機能を内部に持たない組み込み機器が直接 SOAP PRC (Remote Procedure Call)機能を提供する。

2. 組み込み機器における SOAP RPC の検討

2.1 内部実装処理

組み込み機器の典型的な内部実装として、スイッチの制御の場合をとりあげる。以下は、C 言語における実装例である。

```
int controlSwitch (int ControlValue)
{
    // ControlValue に応じたスイッチ制御
    // ...
}
```

本例では、関数 controlSwitch に与える引数の値(例えば 0 または 1)によって、スイッチ状態の変更を行う。

2.2 SOAP RPC と内部実装との対応

前記スイッチを提供する組み込み機器に対して SOAP RPC によって制御を行う場合、機器の内部実装機能を SOAP で規定される XML ドキュメントのタグと要素にマッピングさせなければならない。以下に、前記スイッチ制御関数に対応する SOAP RPC の、属性および名前空間プレフィックスを除く XML ドキュメント部分の例を示す。

```
<SwitchControl>
  <Value>1</Value>
</SwitchControl>
```

なお、内部処理関数の引数や戻り値が C 言語構造体型などで、XML Schema[4]で定義される基本データ型でない場合、シリアライズされた SOAP メッセージは複雑になる。本稿では、XML パーサが搭載できない程度にメモリ容量制限がある組み込み機器の場合、機器にアクセスするためのデータ型は数値や文字列など XML Schema で定義される基本データ型であると仮定し以降の議論の対象とする。

一方、XML ドキュメント内の任意のタグは XPath[5]によって一意に指定することが可能である。また、SOAP メッセージは<Envelope>,<Header>,<Body>要素からなる固定の構造の XML ドキュメントである。これにより、SOAP RPC リクエストの<Body>要素のうち組み込み機器制御メソッドおよび引数に対応する部分を XPath 式で表現することが可能となる。

以上の検討から、組み込み機器外部に XML パーサを配置した場合においても、XPath 式により XML ドキュメント部分を指定することで SOAP リクエストに対応した組み込み機器内部処理を実現可能である。また、上記検討において、通信トランスポート層に依存する点はない。

3. 提案方式と通信シーケンス

前章での検討に基づき、XML パーサ機能を持たない組込み機器が、クライアントにたいして SOAP RPC サーバとして振舞うことを可能とする分散プロトコル処理方式を提案する。

3.1 提案方式を実現するシステム構成例

図 1 に、本稿で提案する分散プロトコル処理による組み込み機器向けシステムの構成例を示す。

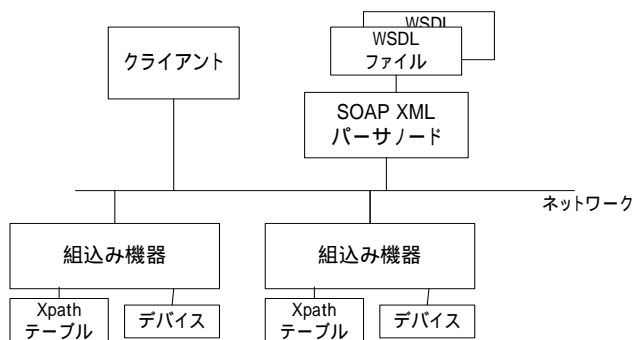


図1: 提案システム構成例

XPath式	関数/引数
/ControlSwitch	*controlSwitch()
/ControlSwitch/Value	SwitchValue

図2: XPathテーブル

本システムは、ネットワークに接続されたクライアント、組込み機器、SOAP XML パーサノード(以下パーサノード)からなる。なお、ここでのネットワークは、IP ネットワークである必要はなく、例えば、RS485 によるシリアル通信路などでも良い。

組み込み機器は、スイッチや温度計など、比較的小規模な監視制御対象としてのデバイスを持つ。これら機器は、クライアントに対する SOAP サーバとして動作する。また、組込み機器は、前節で検討した XPath と内部実装処理とを対応付ける XPath テーブル(図 2)と 1 つ以上の対応するパーサノードのアドレスを持つ。

パーサノードは、組込み機器ごとの WSDL ファイルを持ち、SOAP RPC リクエストの XML ドキュメントの妥当性検証の他、名前空間名やメソッドと引数の対応など検証を行う。

SOAP リクエストを受信した組込み機器は、対応するパーサノードに対して、XML パーシング処理をパーサノードに委託する。これにより、組込み機器内部に XML パーサを持つことなく受信した SOAP リクエストに対応した処理を行うことを可能とする。

3.2 通信シーケンス

以下に、本システムにおける通信シーケンス例(正常ケース)を図 3 に対応して示す。

(1) SOAP リクエストを組込み機器が受信。

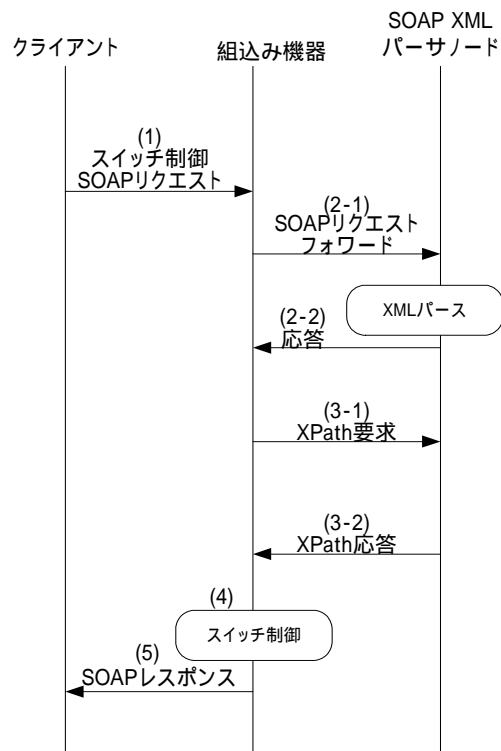


図3: 通信シーケンス例(正常シーケンス)

- (2) 組込み機器は、受信した SOAP リクエストをパーサノードに送付(2-1)。パーサノードでの検証後、応答を組込み機器に返す(2-2)。
- (3) 組込み機器は、XPath テーブルにある XPath 文字列をパーサノードに送付(3-1)、パーサノードは SOAP リクエストの XML ドキュメント内の値を組込み機器に返す(3-2)。
- (4) 組込み機器は、得られた値をもとに対応する内部実装関数を実行。
- (5) 実行結果に対応した SOAP レスポンスをクライアントに返す。なお、Fault 時を含む SOAP レスポンスは XML ドキュメントテンプレートに実行結果を設定することで作成する。

4. おわりに

本稿では、XML パーサ機能を組込み機器外部に配置した分散プロトコル処理方式を提案した。本方式により、組込み機器が XML パーサ機能を持つことなく、SOAP サーバとして機能を提供できる。

参考文献

- [1]W3C Recommendation:Simple Object Access Protocol (SOAP) 1.1 , 24 Jun, 2002
- [2] eSOAP: <http://www.embedding.net/>
- [3] “Web By Proxy”, BILL Gatliff, Embedded Systems Programming, May, 2000
- [4] W3C Recommendation: XML Schema Part2 Data Types, 02 May, 2001
- [5]W3C Recommendation: XML Path Language (XPath) 1.0 , 16 Nov, 1999