

オープンソース開発: Apache SOAP/Axis の開発経験から

中村 祐一

日本アイ・ピー・エム(株) 東京基礎研究所

概要: Linux (<http://www.linux.org/>) が登場して以来, オープンソースによる開発法が注目を集めている. 本稿では, 筆者が係わっている Apache SOAP/Axis というプロジェクトでの体験をもとに, オープンソース開発の利点と問題点を述べる. これらのプロジェクトは SOAP と呼ばれる W3C のスタンダードの処理系を開発することが目的であり, 今後次々と策定されるスタンダードを実装する際のモデルケースとなりえる. 筆者の体験から, オープンソース開発を成功させるには, プロジェクトリーダーを中心とした体制を作り上げることが鍵である, と言うことができる.

Open Source Development: Experience from Apache SOAP and Axis

Yuichi Nakamura

IBM Research, Tokyo Research Laboratory

Abstract: Since the success of Linux, Open Source Development has been receiving special attention all over the software industry. This paper describes its advantages and disadvantages on the basis of experience from Apache SOAP and Axis projects. These projects aim at the development of a processing engine for a W3C standard, SOAP. We could expect that analysis of these projects would be useful for considering projects that address emerging standards. We observe that the key is to organize a good team around a project leader.

1.はじめに

Linux (<http://www.linux.org/>) が登場して以来, オープンソースによる開発法が注目を集めている. オープンソースそのものを厳密に特徴付けるには, ライセンスの形態などに立ち入らなければならないが, ここでは「世界中に分散したプログラマーによってソフトウェアを開発すること」をオープンソース開発と呼ぶことにする. なお, ソースの公開はオープンソース開発に付随するものと見なすことにする.

オープンソース開発がうまく行く理由は既に多くの文献で述べられている. 例えば, 文献[1]では以下のように述べている.

<略> ... おそらく, オープン・ソース・ソフトウェアがこれほどうまく機能する真の理由は, 単に仕事として割り当てられた人々によってではなく, そのソフトウェアに関心を持つ人々によって作成され, 議論が行われるからでしょう. そして, 十分多くの人の目に触れれば, そのソフトウェアのあらゆる部分に対して誰かが関心を持ってくれるでしょう....<略>

このような意見は確かに的を得ているが, 実際のオープンソース開発に係わってみると, 必ずしも利点ばかりではない. 通常, 企業の中では開発チームを作り, ソフトウェアを開発するが, これを開発オープンソースと対照をなすものと位置付け, チーム開発と呼ぶことにする. 容易に想像できるように, 2つの方法にはそれぞれ, 利点と欠点がある.

筆者自身は、もともとチーム開発に係わっていたが、最近になってオープンソース開発に参加するようになった。本稿は、筆者が関わっている Apache SOAP/Axis (<http://xml.apache.org/soap/>, <http://xml.apache.org/axis/>) というプロジェクトでの体験に基づくものである。これらのプロジェクトは現在進行中であり、現状では成功とも失敗とも言えない段階である。しかしながら、筆者は、様々な企業に属する人が共同でソフトウェアを開発することの利点と問題を経験することができたと考えている。以下では、Apache SOAP/Axis の概要、ならびに開発の経緯を説明することにより、そこでの問題点について述べる。さらに、Apache SOAP/Axis の現状に基づいて、オープンソース開発の課題を整理する。

2. Apache SOAP/Axis

SOAP (Simple Object Access Protocol) [2] は非集中/分散環境におけるアプリケーションプログラム間の情報の交換のための XML を用いた単純で軽量なメカニズムを提供する。元々は、XML [3] と HTTP [4] というインターネット上での標準を利用して RPC (Remote Procedure Call) を実現したいという動機から生まれたものである。SOAP 1.1 では、データをモジュールとして符号化するためのパッケージングモデルと符号化のメカニズムを提供することにより、アプリケーションのセマンティクスを表現するための単純なメカニズムを定義している。これによって、RPC だけでなくメッセージングシステムまで非常に広範囲にわたるシステムで SOAP を活用することができる。また、トランスポートからも独立しているので、HTTP だけでなく SMTP [5] や MQ Series [6] 上で SOAP を利用することができる。

SOAP 1.0 はマイクロソフトを中心として策定されたが、1.1 では IBM も加わり、W3C にもサブミットされた。さらに、IBM/MS などが提唱している Web Services 構想の中でも重要な位置をしめるために、大きな注目を集めている。このような重要な技術であるにも係わらず、SOAP のスペックがシンプルであるために、プロトタイプの実成は容易である。このような SOAP の特徴は Apache SOAP/Axis の活動に少なからず影響を与えている。

SOAP 1.1 の発表と同時に、IBM は SOAP4J という処理系を IBM alphaWorks (<http://www.alphaworks.ibm.com/>) 上で発表した。そして、これを Apache に贈呈することにより、Apache SOAP というプロジェクトが立ち上げられた。したがって、Apache SOAP は IBM SOAP4J という既存の処理系に機能を追加するような形で進められており、アーキテクチャ上の大きな変更はなされていない。

一方、Apache SOAP のアーキテクチャそのものが、本来 SOAP が持つ特徴を生かしきれないという議論も盛んに行われていた。その結果、Apache SOAP とは全く違うアーキテクチャをスクラッチから作ることを目的として Axis というプロジェクトが別途立ち上げられた。

オープンソース開発という点からみると、Apache SOAP は Linux ほどではないが、着実に進んでいるように思われる。理由としては、(1) IBM SOAP4J という既存の処理系を基にしている、(2) SOAP4J の開発者がプロジェクトリーダーとして認知されている、といった点が挙げられる。一方で、参加者それぞれ自分の仕事を抱えているためになかなかバグが修正されないことや、ドキュメントがそろわないこともしばしば起こっている。一方で、IBM をはじめ、多くのベンダーで Apache SOAP をバンドルするという動きがあり、Axis にくらべファンシーではないが安定感はある、という評価が一般的である。

これに対して、Apache Axis ではアーキテクチャそのものの設計からメーリングリスト上で議論をはじめたために、半年以上経った現在でも基本的な設計に関する合意も十分に取れていないところがあ

るのが実情である。特に、(1)特定のリーダーとなる人が Apache SOAP ほどはっきりしていない、(2)SOAP そのもののスペックにあいまい性があり、なおかつ XML Protocols (<http://www.w3.org/2000/xml/Group/>) という後継のものも同時に議論されている、(3)SOAP 自体がシンプルな仕様のため様々なレベルの違う人が参加している、といったことが問題となっている。

3. オープンソース開発の課題

筆者の Apache SOAP/Axis の経験のみからオープンソース開発の利点や問題点を総括することはできないが、これらのプロジェクトの現状を分析することはできる。まず、Apache SOAP/Axis は SOAP という W3C のスタンダードの処理系を開発することを目的としている。近年、XML や Web Services (<http://www-106.ibm.com/developerworks/webservices/>) に関するスタンダードが次々と発表されていく中で、今後このようなプロジェクトはますます増えてくると思われる。ところが、たくさんのスタンダードが策定されると、同様の興味を持つ技術者の能力が分散されてしまうことになる。実際、Apache SOAP/Axis に係わる技術者は、両方のプロジェクトに参加し、さらに XML Protocols という新たなプロトコルの策定にも参加しているのが実情である。

最初に、企業内におけるチーム開発とオープンソース開発は好対照のものとして捉えたが、結局良いリーダーがいることがプロジェクト成功のための必要条件である。もちろん、それぞれのプロジェクトで必要とされるリーダーの資質に違いはあるかもしれないが、リードアーキテクト、2 - 3 人のアーキテクト、これらを支えるデザイナー、そしてプログラマ、といった体制がある程度固定できないと、プロジェクトを進められない。オープンソース開発では、特にリーダーが不在の場合、このような体制を組むこと自体に時間がかかってしまうことがある。

最後に、オープンソース開発に係わる企業の意識というものも考えなければならない。オープンソース開発は理念としては興味のある人が集まって特定のソフトウェアを作ることであるが、企業は参加する以上、開発したものを自社製品に組み入れたいと思うのは当然である。できあがったものを組み入れるだけなら問題ないが、企業の戦略にオープンソース開発を合わせたいという願望が入ってくるとプロジェクトそのものが破綻してしまう可能性もある。

参考文献

- [1] Peter H. Salus, “オープン・ソース・ソフトウェアの真のルーツ: 商業的独占ソフトウェアの出現で分かれた進路”, IBM developerWorks 2000. (URL: <http://www-6.ibm.com/jp/developerworks/opensource/roots.html>)
- [2] Simple Object Access Protocol (SOAP) 1.1, W3C Note 08 May 2000. (URL: <http://www.w3.org/TR/SOAP>, 日本語訳: <http://www.trl.ibm.com/projects/xml/SOAP1.1-j-ibm-revision2.html>)
- [3] Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000. (<http://www.w3.org/TR/REC-xml>)
- [4] Hypertext Transfer Protocol -- HTTP/1.1. (<http://www.w3.org/Protocols/rfc2616/rfc2616.html>)
- [5] SIMPLE MAIL TRANSFER PROTOCOL (<http://www.faqs.org/rfcs/rfc821.html>)
- [6] IBM MQ Series Family. (<http://www-4.ibm.com/software/ts/mqseries/>)

オープンソース開発

Apache SOAP/Axisの開発経験から

日本アイ・ピー・エム(株)
東京基礎研究所
中村 祐一

内容

- ◆ オープンソースとは?
- ◆ SOAPとは?
- ◆ オープンソース開発の実際
 - Apache SOAP
 - Apache Axis
- ◆ その他のオープンソース
 - Apache Xerces
- ◆ オープンソース開発の課題
 - Emerging Standards
 - 開発体制
 - 企業の意識

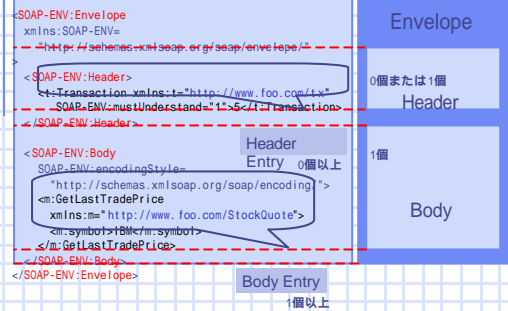
オープンソースとは?

- ◆ Linux
 - 無料
 - 再配布に制限なし
 - ボランティアによる開発(無償で)
 - ◆ オープンソースで一番有名な成功例
- ◆ オープンソース開発
 - ボランティアによる開発の部分にフォーカス
 - Questions?
 - ◆ オープンソース開発は本当に良いのか?
 - ◆ (企業における) チーム開発との違いは?
 - ◆ 何故、企業はオープンソース開発に参加するのか?

SOAPとは?

- ◆ Simple Object Access Protocol 1.1
 - 2000年5月にW3CのNoteとして提案
- ◆ 目的
 - HTTP上でRPCを簡単に実現する(狭義)
 - アプリケーション間でXMLメッセージをやり取りするための枠組みを提供する(広義)
- ◆ 特徴
 - シンプル
 - ◆ XMLメッセージを包み仕様を規定
 - 重要
 - ◆ Web Services 構想 (IBM, MS) での中心的な技術の1つ

SOAP Envelope



HTTPリクエストのメッセージ

```

POST /StockQuote HTTP/1.1
Host: www.stockquotesterver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
  
```

HTTPレスポンスのメッセージ

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
  <m:GetLastTradePriceResponse xmlns:m="Some-URI">
    <Price>34.5</Price>
  </m:GetLastTradePriceResponse >
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

このようなメッセージングは簡単に試せる
e.g. <http://www.xmethods.com/>

SOAP エンジン開発の現状

- ◆ IBM SOAP4J
 - 2000年5月(SOAP 1.1の提出と同時に)対応版がalphaWorksで公開
 - 当初よりオープンソース
 - その後、Apache SOAP プロジェクトに寄付(ASLに)
- ◆ Apache SOAP
 - SOAP4Jをベースに2000年6月ころより活動開始
 - SOAP Attachment, SSL, EJBなどを追加
 - 現在バージョン 2.2
- ◆ Apache Axis
 - SOAP V3として、議論が始まる(2000年8月)
 - 新規アーキテクチャのため独立プロジェクトになる(2000年10月)
 - 2000年7月に1.0リリース予定(目標?)

我々の係わり

- ◆ SOAP Security の標準化
 - SOAPデジタル署名-W3C Note-(丸山, 羽田)
 - SOAP暗号化 活動中(丸山, 羽田, 今村)
- ◆ Apache SOAP
 - コミッター(根山, 中村)
 - IBM WebSphere 4.0 への採用
 - ◆ SOAPデジタル署名の実装(田村, 根山, 中村)
 - ◆ SOAPセキュリティ全般(羽田, 中村)
- ◆ Apache Axis
 - コミッター(根山, 中村)
 - TRL-SOAP の開発を通じて、プロジェクト立ち上げに貢献
 - 現在、セキュリティ担当と認知されている

Apache SOAPの現状

- ◆ 概要
 - アーキテクチャはSOAP4Jと大きく変わっていない
 - 必要なコンポーネントの追加
 - ◆ SSL, SOAP Attachment, Digital Signature...
- ◆ 体制
 - IBM TJW が中心となって活動をリード
 - ◆ Sanjiva (Leader), Matt, Paco, etc.
 - 体制的にも、システム的にも安定
- ◆ 活動状況
 - バグの修正, ドキュメントの整備が十分でない
- ◆ 品質
 - IBM, Oracle などの製品に組み込まれている
 - 設計, 実装は必ずしもきれいでない

Apache Axisの現状

- ◆ 概要
 - SOAPの特徴を生かしたコンポーネント化されたアーキテクチャ
- ◆ 体制
 - IBMが中心となっているが、リーダー不在
 - 体制の不安定さが、設計・開発の迷走につながっている?
- ◆ 活動状況
 - 基本的な設計の部分で合意できてない部分もある
- ◆ 問題点
 - リーダー不在
 - SOAPスペックのあいまい性
 - 色んな人が設計に参加しすぎ?

その他のオープンソース： Apache Xerces

- ◆ 概要
 - IBM XML4J を Apache に寄付して、プロジェクトスタート (田村 @TRL)
- ◆ 体制
 - IBMの1-2人が設計から、実装を担当
 - ソースも開発もオープンだが、実情はオープンソース開発ではない
- ◆ SOAPとの違い
 - XML Parserの開発は多くの人の関心を惹かない (by Andy Clark, @TRL)
- ◆ 問題点
 - 特になし?

オープンソース開発の課題

- ◆ 技術者の不足
 - SOAP・Web Servicesのような新規領域では特に顕著
- ◆ 体制作り
 - 自然発生的に体制ができることを期待しているので、最悪の場合体制ができないままプロジェクトが進行する
 - チーム開発に比べて本当に良いのかは、対象とするソフトウェアによるかもしれない
- ◆ 企業の意識
 - 企業は自社の製品戦略とオープンソース開発を関連付けなければならない。
 - 自社の利益のために、オープンソース開発をコントロールしようとすると、プロジェクトそのものが破綻する

企業における流れ

