Vol.2010-DPS-145 No.27
Vol.2010-GN-77 No.27
Vol.2010-EIP-50 No.27
2010/11/26

RESTful Web　APIによる商品カタログ相互利用サービス

廣岡正史[†]　沓掛裕之[†]　中山真樹[††]　藤井章博[‡]

今後，クラウド上の各種サービス提供の窓口の機能を担うAPIは，オープンな規格が利用されるようになり，標準的な機能と構造を持つAPI同士で企業間のデータ連携が図られると考えられる．筆者らは，オープンな商品Webカタログ専用のCMS(コンテンツ・マネジメント・システム「かた楽」）を開発した．「商品情報」を表現するためのXMLモデルを定義し，RESTfulな実装により，汎用的に利用できるEDIのプラットフォームを実装し，APIとして公開した．

Goods' Catalog Data interchange through RESTful Web API

Masashi HIROOKA[†]　Hiroyuki KUTSUKAKE[†]
Maki NAKAYAMA[††]Akihiro FUJII[‡†]

Recently Open standards of API over cloud environment have been discussed.　API's in which interface of various service are provided will connect different organizations in terms of data linkage such as EDI.　We have developed CMS(contents management system) software called "Cataraku" that provide catalog management feature for goods. XML model for EDI functions are defined and implemented as RESTful API's.

1.　はじめに

クラウドコンピューティング（以下「クラウド」）環境において，実装レベルで何らかの処理を実現するということは，基本的な要素としてプロバイダーによって提供されるAPI（Application Programming Interface）をもとにシステム組み立てることに相当する．現在，クラウドに関連する各種のサービスを提供する大手のASP事業者では，独自に定義したAPIの仕様が利用されている[1]．一方，北米を中心とする有識者によって提言されている「Open Cloud Manifest[2]」によると，今後のクラウドでのAPIの仕様はオープンであるべきであるとしている．こうした考え方をもとに，現在，複数の標準化団体が，クラウドにおけるAPIの標準を検討している[3][4][5][6]．今後，クラウド上の各種サービス提供の窓口の機能を担うAPIは，オープンな規格が利用されるようになり，標準的な機能と構造を持つAPI同士の緩やかな連合という形に落ち着いていくのではないかと考えられる．クラウドが普及すれば，中小企業など対しても高度なIT機能を比較的低コストで提供できる可能性が生まれる．このため，電子商取引のためのEDI(Electronic Data Interchange)等が加速度的に浸透し，幅広い分野でビジネス上の利便性に寄与できる可能性がある．本稿では国際的な標準化の検討の過程に沿って，我々の提案するAPIの詳細仕様について説明しその検討を行う．

2.　研究経緯

筆者らは，機械部品流通に関連する企業とITベンダーによるコンソーシアムを数年前から組織し，APIによって，商品流通に携わる複数企業が比較的容易にEDIを実現する方式について検討を行ってきた．この目的のために，共通に参照される商品マスターのデータベースに対してAPIによりデータの登録・参照・更新・削除（以下「CRUD」）できるプラットフォームを構築した[7]．本稿では，そのプラットフォームを利用して，提示されている商品データに関するEDIを実現するAPI機能について検討する．

最近の研究として例えば，文献[1]では，書誌レコードに関する管理用APIを開発し，図書館の蔵書レコード関連付ける試みが報告されている．本稿では，同様の観点から，RESTful APIを利用し，商品データを取引に関連する複数企業から参照する形でEDIシステムを実現する．以下，オープンな標準化におけるAPIの種別を述べ，開発したAPIの仕様について説明する．さらに，文献[8]で検討したモデルとの関連性を述べる．

[†] (株)山辺　Yamabe Co.
[††]ネクストオブジェクト（株）　Next Objext Co.
[‡]法政大学理工学部応用情報工学科 Hosei Univ., Applied Informatics, Faculty of Science and Engineering

ⓒ2010 Information Processing Society of Japan

Vol.2010-DPS-145 No.27
Vol.2010-GN-77 No.27
Vol.2010-EIP-50 No.27
2010/11/26

3. クラウド API のレベルとカテゴリ

以下では，文献[2]などで進められている検討内容を踏まえて，一般的な立場からクラウド環境で利用される API を「レベル」と「カテゴリ」の観点から分類し，その後の検討の準備とする．

3.1 API のレベル

API を利用する開発者の観点から，API には次の 4 つのレベルの API がある[6].

（1） レベル 1「単純な接続」: もし，API が REST アーキテクチャで実現されているとすると，開発者は適切な HTTP ヘッダを生成する必要がある．これより，API は，HTTP レスポンスの形でデータを返す．REST というシンプルな構造により，開発者は比較的容易にこのレベルの接続機能を利用できる．サービスが SOAP で実装されている場合も同様に適切な SOAP ヘッダを生成し, XML のパーサを利用する. ただし, SOAP で実装されている多くのサービスは以下のより高いレベルの実装となっている.

（2） レベル 2「ツールに依存」: このレベルの API の開発者は，特定のクラウドサービスが提供するツールキットを活用することになる．こうしたツールキットは，REST または SOAP 形式でリクエストを発することになる．開発に際して，接続レベルの検討が必要となるが，例えば，レスポンスやシグニチャの計算などは，ツールキットによって実現できる．

（3） レベル 3「サービスに依存」: 特定のクラウドサービスに依存する開発環境における API．このレベルでは，開発者はビジネスオブジェクトとビジネスプロセスに集中できる．開発者は，接続プロトコルのレベルには関与しないで，組織別に規定されているデータやプロセスに関与するため，その生産性は，高くなる．

（4） レベル 4「サービスに中立」: もっとも高いレベルの API がこれになる．開発者は複数のクラウド環境に対して共通のインタフェースによって実装を行うことができる．もし，クラウドベンダーを変更する場合でも，開発者はその差異を意識する必要がない．

3.2 API のカテゴリ

開発者にとっての API の開発に関するプログラミングは，次の 5 つに分類できる．

（1） カテゴリ 1「通常のプログラミング」: Java，PHP，C#等によるアプリケーションの開発．クラウドに固有の要因を含まない API の開発．

（2） カテゴリ 2「配備用」: クラウド上に何らかのアプリケーションを配備することを前提とした API の開発．WAR ファイルの開発や，.NET のアセンブリなど典型的な Web システムの開発プロセスとなる．

（3） カテゴリ 3「クラウドサービス」: 一定のクラウド環境を前提とした API. レベルの節で分類したように，クラウド上の API は，特定のサービスに特化する場合とサービスに依存しない汎用の API の両方が考えられる．このタイプの API は，さらにストレージ，データベース，メッセージキュー，その他に関連するものとしてさらに細分化できよう．

（4） カテゴリ 4「仮想的な基盤運用」: クラウドの提供する仮想的な計算機資源の運用のための API. 仮想環境上の実装プログラムのアップロードやデプロイ, 停止, 再起動，削除などの操作を行う．基盤運用の対象は，ファイアーウォールや，ノード管理，ネットワーク管理や負荷分散などである．

（5） カテゴリ 5「内部インタフェース」: 異なるクラウド環境の間のインタフェースを取るための API. もし, クラウドを提供するベンダーを変更する場合などには, 新たな API の開発などが必要となる．

クラウド上に企業間 EDI を実施する機能を構築することは､カテゴリ 3 に位置する。

4. 公開 API 仕様

4.1 実装のための API の検討

次に，これらの実装を独自の従来型クライアント・サーバ環境で運用する場合と Google App. Engine などの既存の大手 ASP が提供するパブリッククラウドを利用する場合で比較検討してみる．なお，異なる ASP を跨る実装も考えられるがここでは省略する．

ⓒ2010 Information Processing Society of Japan

Vol.2010-DPS-145 No.27
Vol.2010-GN-77 No.27
Vol.2010-EIP-50 No.27
2010/11/26

・独自の従来型クライアント・サーバ環境（S 環境）
・パブリックなクラウド環境（C 環境）

まず，DB 機能に関しては，蓄積されるデータに関する基本的な CRUD 機能が S 環境でも C 環境でも必要となる．データの受け渡し機能は，RDBS を前提とする S 環境と NoSQL を前提とする C 環境では大きく異なる．しかし，API としてインターフェースの仕様は同一にできるであろう．そこで，実装をレベル 1 の API の範囲で開発することは柔軟性・拡張性観点から意義がある．

次に，EDI 機能の実装に関しては，企業間で取り決められる XML スキーマが共通でなければならない．EDI に必要なデータの受け渡しに関しては，レベル 2，3 の機能を利用する場合は，S 環境と C 環境で実装に大きな差異が生じる．レベル 1 の API で実装されている場合は，C 環境の独自な API 仕様に影響を受けることは少ないと考えられる．EDI 機能を利用する立場からは，特定の C 環境に依存する API でないことが望ましい．そこで，レベル 1 の API の範囲で実装することは，意義があると考えられる．

4.2 システムの全体像

筆者らは，次世代のオープンな商品 Web カタログのありかたを提案する商品カタログ専用の CMS(コンテンツ・マネジメント・システム）として「かた楽（以下「本システム」）」を開発した．レベル 1 の範囲に限定した実装により，S 環境と C 環境共通に利用できる EDI のプラットフォームを実装した．以下では，その実装を説明する．

筆者らは，文献[8]で Cloud_Object という概念の提唱をおこなっている．この考え方は，多様な API のシグネチャを正準化（カノニカル化）して，EDI の標準化に対応しようという試みであり，「レベル 4」に属する．あるデザインパターンにそった API を対外的（＝クラウド側）に提供して，内部的（＝API 提供側）には個々の固有の API に接続するものである．本稿で説明する API は，このようにクラウド環境で機能を提供するための抽象的な API 仕様の実装としての位置づけとなる．

本システムでは，今後，商品カタログ分野について，クラウド環境で標準的な形式が利用されていく考え，「商品情報」を表現するための XML モデルを定義し，公開している．本システムでは，その XML 形式で実際に API の入出力を行っている．公開している仕様は次の 3 点である．以下それぞれの仕様の概要を説明する．

・商品情報モデル仕様　（システムの全体像）
・CMS API 仕様　（DB の CRUD 機能の実装に対応）
・EOS API 仕様　（EDI 機能の実装に対応）



図 1　「かた楽」CMS 管理画面
Figure 1　CRUD management of　“Cataraku”

ⓒ2010 Information Processing Society of Japan

Vol.2010-DPS-145 No.27
Vol.2010-GN-77 No.27
Vol.2010-EIP-50 No.27
2010/11/26

本システムでは，カタログ情報を外部表現する基本構造として，カタログ内の様々な一覧を構成する「一覧要素(element)」と商品内容をあらわす「商品情報」(iteminfo)」という 2 種類の構造を DOM モデル として定義している．Web 画面上ではこれらの商品情報は図 2 のように表示され，一覧要素「element」との具体的な対応関係が示される．また，商品情報 iteminfo の具体例を図 3 に示す．

一覧要素	要素クラス	要素ID	親の要素ID	要素名
	クラスとして可能な分類			
	商品カテゴリー			
	商品			
	外部資料リンク			
	要素説明			
	要素サムネイルURL			
	発注品番	(class="item"の場合のみ)		
	外部資料URL	(class="reference"の場合のみ)		
	子elemetの一覧	(class="itemgroup"の場合のみ)		
		子element		

図 2　商品カタログ内における要素一覧の例
Figure 2　Example of elements in Goods Catalog

4.3 商品情報 DB の API

参照系の API と入力系の API を準備し，DB の CRUD 処理に対応している．

・参照系 API
element (GET 時)
iteminfo (GET 時)
・入力系 API
iteminfo (POST 時)
以下で各項目の説明を行う．

商品情報	商品ID				
	商品発注品番				
	商品ツリー				
		親カテゴリー	親の要素ID	親の要素名	
	アイキャッチ類				
		リード文			
		性能アイコン	画像URL	画像タイトル	
			アイコンの注		
	見出し類				
		商品名			
		商品ロゴ	画像URL	画像タイトル	
		サブ見出し			
		肩見出し			
		カテゴリーマーク	画像URL	画像タイトル	
			アイコンの注		
		商品ラベル画像	画像URL	画像タイトル	
			画像の注		
	別名一覧				
		商品の別名			
	商品概要				
	商品写真一覧				
		商品写真	画像URL	画像タイトル	
			写真の注		
	図の一覧				
		図	画像URL	画像タイトル	
			図の注		
	CADデーター覧				
		CADデータファイル	ファイルURL	データタイトル	ファイルバイト数
			データの注		
	自由項目一覧				
		自由項目	タイトル		
			任意文字列(HTML可能)		
			注		
	欄外注				
	外部リンク一覧				
		外部リンク	リンクの要素ID	タイトル	URL
	スペック表の検索条件				
		検索パラメータ			
	スペック表の行数				
	スペック表				
		表データ			
			見出し行		
				列	
			データ行		
				列	
		スペック表の注			

図 3　XML 定義に基づいた商品情報のレイアウト例
Figure 3　Goods layout based XML Schema

ⓒ2010 Information Processing Society of Japan

Vol.2010-DPS-145 No.27
Vol.2010-GN-77 No.27
Vol.2010-EIP-50 No.27
2010/11/26

4.3.1 element (GET 時)

あるカタログ要素情報（element 要素）を取得するメソッドで，カタログ要素としては
- itemgroup (商品グループ)
- item （商品）
- reference （外部資料へのリンク）

の３種類のクラスを持つ．http://{API エンドポイント}/element/id/{N}/ の形式で，{N}部分で element の id を指定し，その element を表現する XML を取得する．

4.3.2 iteminfo (GET 時)

ある商品情報（iteminfo 要素）を取得するメソッドである．http://{API エンドポイント}/iteminfo/id/{N}/ の形式で，{N}部分で iteminfo の id を指定し，その iteminfo を表現する XML を取得する．また，同じ商品情報を，下記の形式でリクエストすることも可能である．

http://{API エンドポイント}/iteminfo/orderNumber/{XXXXXX}/ の形式で，{XXXXXX}/部分で iteminfo の orderNumber を指定し，その iteminfo を表現する XML を取得する．なお，オプション機能として，このときの戻り値の iteminfo 内に「スペック表」がはいっていた場合，この orderNumber に該当する型番の行だけが返される．

4.3.3 iteminfo (POST 時)

ある商品情報（iteminfo 要素）を新規追加するメソッドである．http://{API エンドポイント}/iteminfo/pid/{N}/ という URL 形式の{N}部分で，この iteminfo を新規追加する対象の親エレメント(=itemgroup)を指定する．そして POST メソッドによって，追加する iteminfo を表現する XML ファイルや画像ファイル類などを送信する．

- 参照系 API
  - element (GET 時)
  - iteminfo (GET 時)
- 入力系 API
  - iteminfo (POST 時)

4.4 EDI の実現のための EOS 機能

EOS は，データという観点からは「価格情報」「受注伝票」などを管理する"データ・リソース"と位置づけられる．本 EOS(公開 1.0 版) においては，これらのデータのうち「価格情報」を API から取得することができる．
本 API では，品番ごとの価格情報を<saleinfo>要素として表現して，API は REST スタイルに準拠している．URL のパスとメソッドによってリクエストが構造化され，XML 形式で値のやりとりを行う．以下で，各 API の仕様を概説する．

4.4.1 saleinfo (GET 時)

これはある saleinfo 要素を取得するメソッドである．http://{API エンドポイント}/saleinfo/orderNumber/{str}/ の形式で，{str}部分で商品発注品番（orderNumber）を指定し，その販売情報（saleinfo）を表現する XML を取得する．取得できる XML の例を以下に示す．

```
<?xml version="1.0" encoding="UTF-8"?>
<catarakuEOS xmlns="http://www.onezero.co.jp/catarakuEOS/ns/" version="1" level="1" >
    <saleinfo orderNumber="neji01" hasPrice="1" needsInquiry="0">
        <label>なべこねじ</label>
        <unit>本</unit>
        <stock>在庫あり</stock>
        <description>販売元備考</description>
        <price>50</price>
        <discounts>
            <discount over="100" price="40" />
            <discount over="1000" price="25" />
        </discounts>
    </saleinfo>
</catarakuEOS>
```


ⓒ2010 Information Processing Society of Japan

Vol.2010-DPS-145 No.27
Vol.2010-GN-77 No.27
Vol.2010-EIP-50 No.27
2010/11/26

●URL 例
URL の例としては，API のエンドポイントが http://www.example.com/catarakuEOS/api/ の場合，例えば「 orderNumber=neji0001」の販売情報を請求する際，次の問い合わせで行う．
http://www.example.com/catarakuEOS/api/saleinfo/orderNumber/neji0001/

●HTTP リクエスト例
GET /catarakuEOS/api/saleinfo/orderNumber/neji0001/ HTTP/1.1
Host: example.com

Authorization:
Digest username="sample", realm="sample", nonce="samplesamplesample"....
この場合，Digest 認証用に Authorization ヘッダを含ませることを前提としてる．

●HTTP レスポンス例 HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 21:19:09 GMT
Server: Apache/2.2.11
Content-Length: 2058
Content-Type: text/xml; charset=utf-8

レスポンス本文は次に示すような XML 文字列となる．

EOS(EDI 機能)の HTTP レスポンス例

HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 21:19:09 GMT
Server: Apache/2.2.11
Content-Length: 2058
Content-Type: text/xml; charset=utf-8

```
<?xml version="1.0" encoding="UTF-8"?>
<catarakuEOS xmlns="http://www.onezero.co.jp/catarakuEOS/ns/" version="1" level="1" >
    <list>

    <saleinfo orderNumber="neji01" hasPrice="1" needsInquiry="0">
        <label>neji01 のラベル</label>
        <unit>neji01 の単位</unit>
        <stock>neji01 の在庫</stock>
        <description>neji01 の備考</description>
        <price>50</price>
        <discounts>
            <discount over="100" price="40" />
            <discount over="1000" price="25" />
        </discounts>
    </saleinfo>

    <saleinfo orderNumber="neji02" hasPrice="1" needsInquiry="0">
        <label>neji02 のラベル</label>
        <unit>neji02 の単位</unit>
        <stock>neji02 の在庫</stock>
        <description>neji02 の備考</description>
        <price>50</price>
        <discounts>
            <discount over="100" price="40" />
            <discount over="1000" price="25" />
        </discounts>
    </saleinfo>

    <saleinfo orderNumber="neji03" hasPrice="1" needsInquiry="0">
        <label>neji03 のラベル</label>
        <unit>neji03 の単位</unit>
        <stock>neji03 の在庫</stock>
        <description>neji03 の備考</description>
        <price>50</price>
        <discounts>
            <discount over="100" price="40" />
            <discount over="1000" price="25" />
        </discounts>
    </saleinfo>

    </list>
</catarakuEOS>
```


ⓒ2010 Information Processing Society of Japan

Vol.2010-DPS-145 No.27
Vol.2010-GN-77 No.27
Vol.2010-EIP-50 No.27
2010/11/26

●エラー時
エラーの際は，下記の形式のエラーメッセージが戻る．

```
<?xml version="1.0" encoding="UTF-8"?>
<catarakuEOS xmlns="http://www.onezero.co.jp/cataraku/ns/" version="1" level="1" >
<error>エラーメッセージ</error>
</catarakuEOS>
```

4.4.2 saleinfo_list

これは，API を利用して，本システムが提供するデータとのマッシュアップによる連携を行う場合に，saleinfo 要素を配列で取得するメソッドである．http://{API エンドポイント}/saleinfo_list/ に対して商品発注品番（orderNumber）をスラッシュ区切りで並べたリクエストを POST して，その販売情報（saleinfo）のリストを表現する XML を取得する．取得できる XML はこの形式における<saleinfo>タグが， <list>内に列挙された形式になる．レスポンス例を以下に示す．

```
HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 21:19:09 GMT
Server: Apache/2.2.11
Content-Length: 2058
Content-Type: text/xml; charset=utf-8

<?xml version="1.0" encoding="UTF-8"?>
<catarakuEOS xmlns="http://www.onezero.co.jp/catarakuEOS/ns/" version="1" level="1" >
    <list>

    <saleinfo orderNumber="neji01" hasPrice="1" needsInquiry="0">
        <label>neji01 のラベル</label>
        <unit>neji01 の単位</unit>
        <stock>neji01 の在庫</stock>
        <description>neji01 の備考</description>
        <price>50</price>
        <discounts>
            <discount over="100" price="40" />
            <discount over="1000" price="25" />
        </discounts>
    </saleinfo>

    <saleinfo orderNumber="neji02" hasPrice="1" needsInquiry="0">
        <label>neji02 のラベル</label>
        <unit>neji02 の単位</unit>
        <stock>neji02 の在庫</stock>
        <description>neji02 の備考</description>
        <price>50</price>
        <discounts>
            <discount over="100" price="40" />
            <discount over="1000" price="25" />
        </discounts>
    </saleinfo>

    <saleinfo orderNumber="neji03" hasPrice="1" needsInquiry="0">
        <label>neji03 のラベル</label>
        <unit>neji03 の単位</unit>
        <stock>neji03 の在庫</stock>
        <description>neji03 の備考</description>
        <price>50</price>
        <discounts>
            <discount over="100" price="40" />
            <discount over="1000" price="25" />
        </discounts>
    </saleinfo>

    </list>
</catarakuEOS>
```


ⓒ2010 Information Processing Society of Japan

Vol.2010-DPS-145 No.27
Vol.2010-GN-77 No.27
Vol.2010-EIP-50 No.27
2010/11/26

5. むすび

現在，クラウドによる企業情報システムの構築が盛んに議論されている．パブリック，プライベートの別にかかわらず，企業がオープンな規格に基づいてクラウド上の EDI を導入する場合，の実践に関してもたらす実務上の変化は，ホスティングに必要なハードウエアの設置が仮想環境に移行すること，データ交換のためのソフトウエアがクラウド上で実行されるために事実上共通化できる点である．

EDI の普及を困難にしている要因は四つ挙げられる．第一は，システム導入の複雑さである．第二の要因は，セキュリティに関する問題で，識別，秘密性，整合性，認証及び否認防止に関連する内容である．第三の要因は，標準化及び／又は標準化の普及の不十分さである．第四の要因は，契約に関わる商法上問題と，電子文書の証拠としての価値・評価に関する問題である．本研究は，第三の要因に焦点を当てているが，第一と第二の要因についても，クラウドによって緩和される可能性がある．第四の要因に関しては，本研究では取り扱っていない．

クラウドの今後の普及の可能性を考えると EDI の導入を意識して，レベル１の範囲で API を準備し，公開することは意義があると考える．こうした取り組みが普及することによって，現在の Web ベースの EDI の問題点である「多画面現象」を改善できると考えている．今後の研究課題は，実践的な環境での開発システムの評価である．

謝辞　法政大学理工学部応用情報工学科クラウド実装研究開発プロジェクト関連企業の関係者の皆様(www.onezero.co.jp)に感謝申し上げます．

参考文献

1)]安藤友晴「分散システムで FRBR モデルの書誌レコードを管理する RESTful API の設計と利用」情報処理学会研究報告，Vol.2009-FI-96NO.7(2009/11/19)
2) Open Manifesto: http://www.opencloudmanifesto.org/Open%20Cloud%20Manifesto.pdf
3) “Results of Clouds SDO Survey”,
IETE2010.6http://www.ietf.org/mail-archive/web/clouds/current/pdfkrmoPVEfqP.pdf
4) OGF: http://www.gridforum.org/
5) SNIA: http://www.snia.org/
6) DMTF: http://www.dmtf.org/
7) Cataraku: http://www.onezero.co.jp/
8) 藤井章博他,「Web API を活用した EDI 機能の検討」 情報処理学会研究報告, EIP48-9, 2010 年 5 月
9) 藤井章博,「広がる Web API の活用―マッシュアップの幅広い可能性―」, 科学技術動向 No.106, 文部科学省科学技術政策研究所,2010 年 1 月

ⓒ2010 Information Processing Society of Japan