Vol.2012-DPS-152 No.19
Vol.2012-GN-85 No.19
Vol.2012-EIP-57 No.19
2012/9/14

SVG をリソースとして活用するシステムの検討

吉田弘輝[†1]　藤井章博[†2]

従来, CAD 図面の運用機能は限られたニーズに対応するプロプライエタリーなアプリケーションソフトウエアによって提供される場合がほとんどである．Web 技術の進展により多様な Web サービスが，共通の運用基盤，すなわち Web ブラウザと HTTP プロトコルによって提供できるようになっている. 特に, 最近 Web ブラウザに導入が進んだ HTML5 の canvas 機能により SVG 形式の画像情報をタブレットや PC 上に柔軟に利活用することが可能になった．本研究では，機械部品の CAD 図面等を基に作成した SVG データをリソースとして蓄積し，RESTful なアクセス機能を実装したデータベースシステムを構築した．また，こうしたリソースを利用して多様な電子商取引のニーズに対応するためのマッシュアップ機能を備えたミドルウエアの開発を行っている．機械部品流通などの分野における EDI に対するニーズは多岐にわたる．本稿では，開発中のシステムを利用して機械のメンテナンスのために部品の在庫管理を行うシステムの実装を述べ，特に SVG の利活用に関する課題を検討する．

A RESTful SVG Resource Management System and Its Mashup Applications

HIROKI YOSHIDA[†1]　AKIHIRO FUJII[†2]

When CAD data is utilized in a certain business work flow, the management of the data is usually done by associated proprietary data management software/system. Thanks to the rapid advancement of Web technologies, variety of applications could be designed based on common software platform, namely Web and HTTP procedures. Especially, HTML5 and canvas features which is recently introduced to common browsing software provides huge potential for CAD data applications since SVG format that could be produced from original CAD data, is able to be treated over Web browsers. In our research group, constructions of CAD database with SVG format is going on. Those resources are stored in RESTful architecture that makes it easy for applications to access and utilize them. At the same time, middleware for mashing up those resources are also being designed. We believe this type of system could be useful in various EDI(Electronic Data Interchange) needs. In this article, an implementation of mechanical parts inventory management system is shown as an example of our resource management and mashup middleware.

1. はじめに

CAD（Computer Aided Design）による図面データは，多様な業務で利用されている．一方，Web を介した各種サービス提供方法は日々進化しており，タブレット型の携帯端末の利用が急速な広がりを見せる中, CAD データの表示用に活用し業務支援を行うシステムも登場している．

本稿では，Web サービスとして CAD データを蓄積したドキュメントデータベースを利活用するシステムの提案を行う．CAD 図面を SVG（Scalable Vector Graphics）フォーマットでドキュメントベースに蓄積し，ドキュメントを構成する DOM（Document Object Model）を対象とし，この情報リソースを体系的に管理運用する．情報の管理運用のために，これを RESTful な Web API （Application Programming Interface） によって参照可能とするような実装を行うことでマッシュアップなどの開発手法を通じて多様なアプリケーションで利活用できる．

本稿の構成は，まず 2 で SVG の概要を述べる．次に，3 で SVG など XML 形式のリソース管理システムに関する従来研究を概観する．4 では，提案する運用形態を実装したシステムの解説を行う．5 で考察を述べる．

2. SVG の概要

SVG とはベクターグラフィックスの 1 つであり，XML で記述された形式の画像ファイルであり，W3C によって 2001 年 9 月に SVG1.0 として勧告され 2011 年 8 月に勧告された SVG1.1（Second Edition）まで 5 回の勧告が行われている．

SVG は 2 つの画像データとして利活用するための重要な特徴を持ち, 第一は SVG がベクター形式であるという点である．ベクター形式の画像は，画像を点の座標とそれを結ぶ線の方程式で表現する．そのため，画面を表示する都度計算を行って最適な形での画像の表示ができるため，拡大や縮小といった処理を行っても画像の劣化なく表示することができる．これは，写真などの画像で利用される JPEG や PNG などのラスターグラフィックスにはない利点である．SVG の欠点としては，ベクター画像が線や図形の集合とその図形や線に囲まれた部分に対して色彩情報を付与して表現するため，写真のような色の変化が複雑な画像に対してデータ量が増大するという問題がある．

さらに重要なもうひとつの特徴は，SVG が XML 形式で記述されたデータとして扱えるという点である．つまり画

Vol.2012-DPS-152 No.19
Vol.2012-GN-85 No.19
Vol.2012-EIP-57 No.19
2012/9/14

像でありながら，テキストを扱う要素技術である，HTML や HTTP と親和性を有する．そのため，DOM を用いたプログラムで SVG を操作することができる． これによって画像をインターネット上などで動的に扱うことを可能にすることができる．例えば，画像の一部分だけを変化させたい際に，SVG ならば DOM のプログラムを準備するだけで実現することができる．つまり，すべての SVG は要素や属性といったものだけで表現され，それらの追加・修正・削除を行うことで画像の操作ができる．

特に，2008 年に W3C によってドラフトが発表された HTML5[5]では，Canvas 機能が採用され，SVG をグラフィックスとして表示するための機能が充実した．このことによって，WYSWYG として本来の CAD 図面の具備する視覚表現性と DOM 構造を有する XML 文書として操作性の双方を活用することができるようになった．本研究の動機は，こうした背景のもとでより柔軟に SVG を活用するためのシステムの検討である．

3. 先行研究

CAD（ Computer Aided Design：コンピュータ支援設計）は，半導体製造など幅広い分野で利用されている．90 年代から CAD データの運用と蓄積に関する研究が行われており，例えば文献[1]では，CAD 利用の黎明期に分散環境で CAD データを運用する方式が示された．文献[2][3] では，XML 文書の構造を反映したメタデータを対象として検索する手法が示されており，構造化文書の取り扱いを DOM を介して実施することの有効性を示す一例となっている．

近年，クラウドコンピューティング環境の急速な浸透に伴って，文書レポジトリに蓄積された情報資源をマッシュアップなどの考え方に基づいて活用する試みが増大している．こうした状況を背景とした構造化文書運用管理・運用の在り方も盛んに研究されており，例えば文献[9]では，クラウド環境で SaaS（Software as a Service）の形態によって文書の運用サービスを提供することを前提としたシステムのアーキテクチャを提案している．また，文献[4]では，具体的にアパレル分野で利用される型紙に対応した XML を管理・運用する方式の検討がなされている．

特に，ROA（Resource Oriented Architecture）の考え方のもとで，各種の情報リソースの運用方法を検討する研究は，近年活発に行われている．RESTful なドキュメントデータベースは，その「べき等性（idempotence）」という特徴ゆえに，動的なプロセスに利用することは不向きであるのではないかという危惧がある．こうした懸念に対して，例えば，文献[11]では，RESTful なデータベースから提供されるリソースを利用してビジネスプロセスを実行する場合にも，セマンティックウェッブの機能を適用させることで，リソースの動的な利用に関しても有効に機能することを示している．

4. SVG 利用システム

本稿では，CAD 図面を基に生成された SVG リソースに対して，それが管理運用上の意味のある一定の構造を有している場合に「SVG-DOM」と呼ぶことにする．XML 文書が一定の構造を持っているかどうかは，明確に規定できる特性である．しかし，CAD 図面の場合は，作成された段階で，図面に描かれた構造物のモデルは必ずしも厳密に定まっているわけでない．本稿では，CAD 図面から SVG データを生成過程で，実務的な運用上の要求を満たす範囲で構造化を付与することとする．以下では，まず，「SVG-DOM の生成」について述べ，その利用に際して RESTful なデータベースを構成する部分とこれをマッシュアップによって利用し EDI（Electronic Data Interchange）に利用する．

4.1 SVG-DOM の生成

CAD 図面は，一般的に専用ソフトウエアで作成される．専用ソフトウエアはそれぞれ独自のデータ形式を持つが，業界で共通に利用できる dxf 形式によって相互運用が可能である．Inkscape 等のソフトウエアを利用すると，dxf 形式から SVG 形式を生成することができる．この過程は，CAD ソフトで図面を生成する際の図面の一部，例えばエンジンが全体であれば，その一部としてのキャブレターが図面の中でも一つの塊として表現される．CAD の操作上は，レイヤーやグループ化などによって部分の構造化がされている．一般的には，既存の図面が必ずしも構造を意識しているとは限らないため，SVG を構成する段階で一定の構造化を意識して，XML ドキュメントを編集する必要がある．

以下，図表 1 として SVG-DOM を CAD データから生成する流れをファイル形式とそのファイル形式で利用することができるアプリケーションソフトを元に示す．図表 1 に示したように CAD の共通形式である dxf ファイルは Auto CAD 等のソフトで閲覧などを行うものである．その dxf 形式のファイルを inkscape 等の変換ソフトで svg ファイルに変換し，XML エディタを用いてマッシュアップをおこなうために必要な準備を行うことで SVG-DOM とすることができる．

図表 1　SVG-DOM 生成に関連するデータ

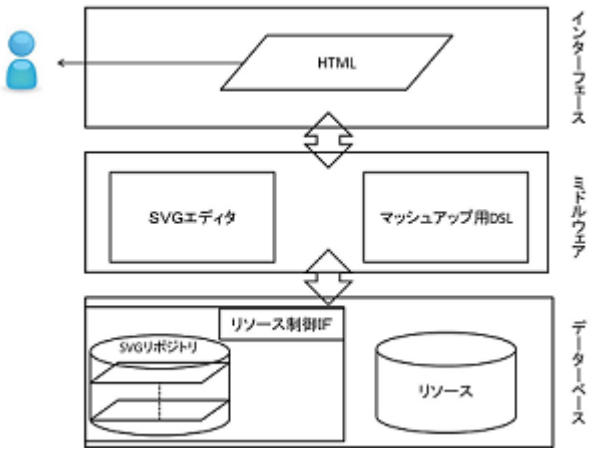
データの構造	ファイル形式	アプリケーションソフト
CAD　共通形式	.dxf	Auto CAD 等
dxf 形式から変換	.svg	Inkscape 等
SVG-DOM	.svg	XML エディタ等

4.2 SVG-DOM の利用モデル

我々は，XML ドキュメントをリソースとして運用する際，RESTful なインターフェースによって実装することで，柔軟で発展性のある情報システムが構成できると考えている．

Vol.2012-DPS-152 No.19
Vol.2012-GN-85 No.19
Vol.2012-EIP-57 No.19
2012/9/14

SVG をリソースとし, Web 上で運用するためのドキュメントベースは，インターフェース，ミドルウェア，データベースといった 3 つの階層に分けて考えることができる．図表 2 に 3 つの階層に分けたモデル図を示す．



図表 2　SVG を利用するための 3 層モデル

4.2.1　リソース・データベース（最下層）

システム構成の最下層であるデータベースは複数の SVG などのリソースを保持する階層である．この階層は，クラウドコンピューティング環境での, NoNSQL 型の XML データ向けのストレージを利用する．

図面はメタ情報と共に蓄積される. CAD 情報のリソースである SVG は他の XML などと組み合わせる．

これらの情報リソースに対して RESTful な Web API（Application Programming Interface） によって実装することによって SVG を効率良く利用する環境を構築することができる．また，SVG と組み合わせるためのデータは，インターネットを介してネットワーク上のどこかにあるリソースを利用することでマッシュアップをおこなうことも可能である．

4.2.2 ミドルウェア

システム構成の中間に位置するミドルウェアは，SVG のエディタ機能とマッシュアップ用の DSL（Domain Specific Language）の 2 つの機能を有している．ひとつめの機能である SVG エディタは SVG のリポジトリからから取得した SVG ファイルを使用目的に合わせて SVG のオブジェクトである DOM 毎に加工を行う．また，他にも SVG ファイルにリンクなどを作成することで今後のマッシュアップを円滑に行うための準備を行うなどの処理を行うためのものである．

次にもうひとつの機能であるマッシュアップ用の DSL であるが，DSL とはドメイン特化言語のことであり，特定のタスク向けに設計されたコンピュータ言語を意味する．本稿では，SVG の DOM を操作し EDI を実装するために開発した DSL の一部を開設する．本研究では，実装例として JavaScript を利用して関数を定義し, その関数名を要素として DSL の構築を行う．

4.2.3 インターフェース

システム構成の最上層はインターフェースとして加工した SVG などの情報リソースを HTML ファイルとしてユーザが閲覧することができるようにするものである．HTML ファイルに加工することで Web ブラウザを用いてデータを利用することができるようになるため，動画や音声などといった多種多様なリソースを柔軟かつ効率良く利用することができるようになる．また，HTML ベースに情報リソースを行うことでスマートフォンやタブレット PC などの複数の環境でユーザが情報を利用することができるようになるという利点もある．

5.　実装例

本研究で試作する図面用動的情報表示機能を具備したシステムは，クラウド上の DXF などのデータ形式の CAD 図面から変換した SVG の画像に対して複数の処理を行うことで紙媒体の図面以上の情報を提供することを可能にするものである．このシステムには図面に対して XML で記述された部品表をマッシュアップする機能が具備されている．

5.1　データベースシステムの実装

今回試作したシステムで SVG やマッシュアップを行う情報リソースを保持するデータベースとして Google が提供するクラウド環境上のプラットフォームである GAE（Google App Engine）を利用することで簡易的なクラウド上のデータベースとした．これは，本研究の本質である DSL を用いた SVG と情報リソースのマッシュアップに関するシステムの開発に重点を置くためである．

5.2　ミドルウエアの実装

今回試作した DSL は SVG を動的に取り扱うためのものと，SVG と情報リソースをマッシュアップするためのもの, マッシュアップの準備を行うためのものの 3 種類に分類することができる．また，DSL は抽象化されているため，全ての SVG-DOM に対して一定の動作を約束するものである．図表 3 に本研究でシステムのために検討された DSL の一覧を SVG 操作，マッシュアップ，準備の 3 つに分類したものの一覧を示す．

まず，SVG の操作とはリソースとしての SVG を対象としたデータのやり取りや SVG を HTML ファイルで利用するための加工といった処理について記述した DSL のことである．次に，マッシュアップを行うための DSL はブラウ

ザに表示されている SVG と対応する情報リソースをブラウザで表示するなどのマッシュアップを行うための処理についての記述がなされている．最後に準備という項目の DSL は SVG のリストやマッシュアップの対象のリストといったマッシュアップや SVG には直接的には関連していない処理についての記述がなされているものである．

図表 3　試作 DSL 一覧

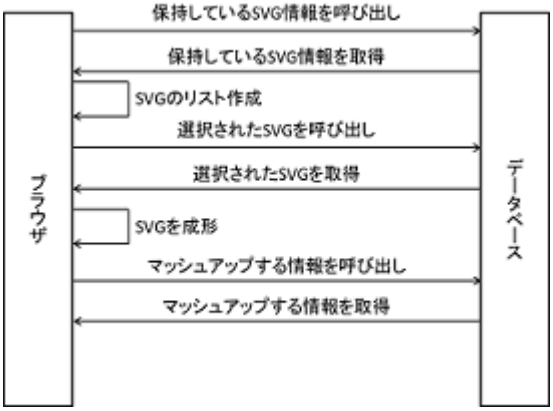
名前	分類	動作内容
Make_SVG_List	準備	DB 上の SVG のリストの作成
Make_data_List	準備	SVG とマッシュアップを行う情報リソースのリストを作成
Get_SVG	SVG 操作	クラウド上の DB から目的の SVG を取得する
Display_SVG	SVG 操作	Get_SVG で取得した SVG をブラウザ上で表示する．
Edit_SVG	SVG 操作	マッシュアップを行うために DB 上の SVG データの加工を行う
Get_Mashup_data	マッシュアップ	マッシュアップに使用する情報リソースを取得する
Display_Mashup_data	マッシュアップ	取得した情報リソースをブラウザで表示する形式に成形する．

また，一例として部品表とのマッシュアップを行うため 2 つの DSL 流れを示すことで DSL が JavaScript でどのような流れで記述されているかを示す．

```
Get_Mashup_data(name){
        get_xml（今回は XML 形式のデータを利用するので XMLHttpRequest で XML を取得する）
        change_DSL（異なる DSL を利用する．この場合は Display_Mashup_data に処理を移行する）
}

Display_Mashup_data(){
        Make_table（取得した XML データをユーザが利用しやすいように表の形に成形する）
}
```

上記した DSL は変数 name に対応した XML ファイルを非同期通信で取得して表の形に成形したものである．このように，JavaScript の関数で記述された DSL をファイルに対して定義するのでなく，目的別に定義することで抽象化することによってすべての SVG-DOM は特定の方式によって情報リソースをユーザに提供するように設計されている．また，上記した DSL の実際の流れに関しては本稿の付録として実際のプログラムを一部記述する．図表 4 に試作したシステムが SVG などの情報リソースを提供するさいの流れを示す．



図表 4　EDI 処理の流れ

5.3 インターフェースの実装

本研究で試作したシステムの目的は複数の SVG と XML で記述された情報リソースを動的にマッシュアップさせることで図面により多くの情報を付与すると同時に，SVG を柔軟かつ効率良く利活用することの 2 つを目的としている．また，利用環境として PC のみでなくスマートフォンやタブレット PC も想定し，検証を行った．図表 5 に検証を行ったタブレット PC の環境について示す．

図表 5　UI 実装環境

ブラウザ	Chrome/Opera/Safari/Firefox
OS	Android 2.2/3.0
ハードウェア	Galaxy TAB/Optimus

開発を開始した 2010 年の 12 月の段階では Galaxy TAB を利用した Android のバージョンは，2.2 であったため，その上で動作する SVG に対応するブラウザは限られていた．そのため，PC 環境での利用をそう手下開発を先行させた．そして，2011 年 5 月に国内で Android3.0 搭載機種（Optimus 等）が販売され SVG の DOM 操作が機能するブラウザが対応するようになった．

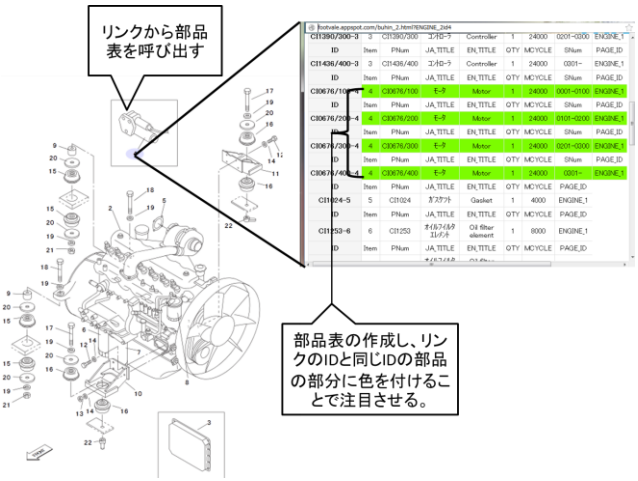
試作システムはインターフェースの実装に JavaScript を利用した．今回 JavaScript を利用した理由は，実際に DOM を適用する HTML ファイルが動作する環境である Web ブラウザ上で操作に適しているプログラミング言語ひとつだからである．

5.4 動作例

試作したシステムとして，最も重要な機能は実際に SVG

Vol.2012-DPS-152 No.19
Vol.2012-GN-85 No.19
Vol.2012-EIP-57 No.19
2012/9/14

と XML で記述された部品表をマッシュアップする機能である. この機能は SVG に付与したリンクタグから部品表を表示するための HTML を呼び出すものである. この際部品表はリンクタグに割り振られた ID と部品表の ID 部分が等しい部分を特定し，その部分を色分けすることができる. 図表 6 としてこの動作を行った際の様子を示す.



図表 6　SVG と部品表のマッシュアップ例

6. 考察

6.1 ビジネスモデル

2 章で述べたように SVG には他の画像フォーマットとは異なる特徴があり，その特徴を生かすことでビジネス・教育・医療・地域社会などで活用できる.

本稿で取り上げた事例は，建設重機を製造するメーカが保有する CAD 情報を重機の保守事業者が重機ユーザのために活用するものである．実際には，その間にリース事業者などが介在するが, 少なくともこのビジネスモデルには，上記 3 つのプレーヤが存在する．リソースとしての CAD 情報はもともと重機メーカが保有する．実装したメンテナンス用の在庫部品とのデータ連携は，保守事業者のニーズである．従来のクライアントサーバ型のプロプライエタリなアプリケーションでは, CAD 情報というリソースをこのニーズに活用することは困難である．RESTful なデータベースの構築と Web API を介したアクセスが可能であれば，本稿で示すようなビジネスニーズに即したマッシュアップにより柔軟で新しいサービスを提供できる.

6.2 システムの機能

つぎに，SVG を活用する観点から検討する．活用の方法は大別して 2 つある．第一の方法は，SVG が持つ拡大縮小しても画像の劣化が起きない特徴を生かした正確な画像を利用したサービスを行うことである．精密部品の図面などの画像を SVG として表現することで, より使い勝手がよい図面として利用することができる. もう一つの方法は, SVG と XML などの情報リソースをマッシュアップすることで画像に多くの情報を持たせることができる. 特に SVG の要素の一つ一つが情報リソースを含んでいるオブジェクトとみなすことができるため，他の画像ファイルより情報リソースとのマッシュアップに向いている.

マッシュアップに適した SVG-DOM を生成する上でマッシュアップを行うためのリンクタグの作成は重要な事柄である. 一般的に SVG 用いられる図面は dxf 形式のファイルから変換ソフトを用いて生成される．こうして作成された SVG にはリンクタグが作成されていない. そのため HTML をインターフェースとしてシステムを構築する場合リンクタグを SVG に対して作成する機能を必要とする. この機能は 4 章で示したミドルウェア内の SVG エディタとして完備させることが良い．この際，クリックなどでユーザが感覚的に作成する座標を決定できるようにすることが重要である.

また, 本研究で試作したように HTML をベースに情報リソースのやり取りを行うことで，従来の PC のみでなくスマートフォンやタブレット PC を用いて情報リソースの利活用を行うことができる．これにより，従来の PC を持ち込むのが困難だった状況での情報リソースの利活用が可能になる．また，HTML は端末にブラウザさえ導入されていれば閲覧することが可能なので，情報リソースを利用するためのハードルを下げることができる.

SVG を利用した EDI を行う上で重要なこととして SVG 内のオブジェクトである一つ一つの要素に統一的な ID をつけることである．ID をつけることで SVG の要素とマッシュアップを行う情報リソースとの関係性を明確にすることができるため，SVG の管理を容易にするため，結果としてより有用なサービスとすることができる.

7. むすび

SVG-DOM を活用することで，本稿で示したように，他の情報リソースとのマッシュアップを行うことで単なる図面としてのリソースが，多様なビジネスアプリケーションにも活用できることが示唆されている. 今後 Web サービスとして SVG を利用していく上で RESTful な環境やクラウドコンピューティングによるレポジトリといったものを活用捨てシステムを構築していくことで Web のようなスケーラビリティやユーザビリティを持たせることができる.

本研究の今後の課題としてはマッシュアップや SVG の加工などを行う DSL の構成や SVG エディタとしての機能といったミドルウェアにと，RESTful な環境における SVG の取り扱い方などのシステム全体の構成といった 2 の事柄について研究を進めることでより柔軟かつ効率良く SVG を利用できるシステムについて検討していきたい.

謝辞

本研究の実装に際して，資金援助と具体的なビジネス事例

を提供いただいた（株）エディエステクノロジー社にこの場をお借りして深謝申し上げます.

参考文献

[1] G. Yamamoto, R. Kondo, S. Hara, M. Oshima, ”A Distributed CAS System for Control System Design”, IEEE IECON’91, 1991 年
[2] 井形伸之，難波功，「大規模な構造化文書データベースにおけるイデクシングと検索の手法」情報処理学会　情報学基礎研究会・自然言語処理研究会資料，2000 年 3 月
[3] S. H. Maes, et al.,“A DOM-based MVC Multi-modal e-Business”, IEEE Int’ Conf” on Multimedia and Expo., 2001 年
[4] Yinglin Li, et. al, “Research on Apparel CAD Pattern Data Interchange Format Based on XML”, World Congress on Software Engineering, 2009 年
[5] W3C, “HTML5: A vocabulary and associated APIs for HTML and XHTML” http://www.w3.org/TR/html5/

[6] エディエステクノロジー（株） http://www.ads-techno.co.jp/
[7]山本陽平，「Web を支える技術」技術評論社 2011 年
[8]Bill Burke，菅野良二「Java による RESTful システム構築」オライリージャパン，2010 年
[9]E.M.Maximilien et al, ”An Online Platform for Web APIs and Service Mashups”, IEEE Internet Computing, 2008
[10]Jim Yu et. al, “Understanding Mashup Development”, IEEE Internet Computing, 2008
[11] Xiwei Xu, etal, “Resouce-Oriented Architecture for Business Processes”, Proceedings of 15th Asia-Pacific Software Engineering Conference, 2008
[12]Leonard Richardson, Sam Ruby，山本陽平監訳，「RESTful Web サービス」，オライリージャパン，2007 年
[13] Debasish Ghosh （著），佐藤竜一 （監修，翻訳），「実践プログラミング DSL ドメイン特化言語の設計と実装のノウハウ」2012 年 6 月，翔泳社

附録（マッシュアップ　JavaScript）

以下の JavaScript のコードは本稿 5.2 章で記述した SVG と部品表のマッシュアップを行うための DSL を具体的に示したものである．以下のコードは引数である name で部品表を特定することで, XML ファイルを呼び出す. そして，呼び出した XML ファイルの中身をユーザが閲覧しやすいように表の形になるように HTML ファイルのタグを生成している．

```
function hyou(name){
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.open("GET", "./XML/"+ name +"_1.xml", false);
        xmlhttp.send(null);

        var xmlDoc = xmlhttp.responseXML.documentElement;
            var part = xmlDoc.getElementsByTagName("Part");
            var df = document.createDocumentFragment();

            for(var i = 0; i< part.length;i++){
              var pTag = document.createElement("tr");
              var p2Tag = document.createElement("tr");
              var aTag = document.createElement("th");
              var a1Tag = document.createElement("th");

        a1Tag.appendChild(document.createTextNode("ID"));
                p2Tag.appendChild(a1Tag);
                set(part.item(i),p2Tag);
                df.appendChild(p2Tag);

        aTag.appendChild(document.createTextNode(part.item(i)
.getAttribute("ID")));
                            pTag.appendChild(aTag);
                            start(part.item(i),pTag);
                            df.appendChild(pTag);
            }

        document.getElementById("result_2").appendChild(df);
        }
```