

SuperSQL による HTML5 の生成

石崎 文規[†] 遠山元道^{††}

[†] 慶應義塾大学理工学部情報工学科 〒 223-8522 神奈川県横浜市港北区日吉 3-14-1
E-mail: †ishizaki@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

あらまし SuperSQL とは関係データベースの出力結果を構造化し、多様なレイアウト表現を可能とする SQL の拡張言語である。本研究においては SuperSQL において従来の HTML 出力では対応していない HTML5 の様々な機能に対応した出力を行うための実装を行った。

キーワード SuperSQL, HTML, HTML5, 問い合わせ言語

Generation of HTML5 in superSQL

Fuminori ISHIZAKI[†] and Motomichi TOYAMA^{††}

[†] ^{††}Department of Information and Computer Science ,
Keio University

Hiyoshi 3-14-1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223-8522 Japan
E-mail: †ishizaki@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

1. はじめに

近年、次世代のウェブ環境に合わせて、新たに仕様を検討されているのが HTML5 であり、HTML 言語の最新バージョンとして、現在標準化作業が進められているので、数年後の Web ページは HTML5 で作られたものが主流となる事が予測される。

SuperSQL では HTML 出力に対応しているが、HTML5 の新しい機能を使った出力は行われていない。そこで HTML5 の新しい機能を出力出来る仕組みを SuperSQL に組み込み、SuperSQL の HTML5 対応を提案する。

以下、本稿の構成を示す。まず 2 章で SuperSQL の概要を述べ、3 章で SuperSQL における HTML5 への対応について述べる。次に 4 章で HTML5 対応への実装について述べ、そして 5 章では評価について述べ、6 章で結論およびまとめを述べる。

2. SuperSQL

この章では本論文で改善を試みる SuperSQL について簡単に述べる。SuperSQL は関係データベースの出力結果を構造化し、多様なレイアウト表現を可能とする SQL の拡張言語であり、慶應義塾大学遠山研究室で開発されている [1] [2]。そのクエリは SQL の SELECT 句を GENERATE< media >< TFE > の構文を持つ GENERATE 句で置き換えたものである。ここで < media > は出力媒体を示し、HTML, PDF などの指定ができる。また < TFE > はターゲットリストの拡張である Target Form Expression を表し、結合子、反復子などのレイアウト指

定演算子を持つ一種の式である。

2.1 結合子

結合子はデータベースから得られたデータをどの方向 (次元) に結合するかを指定する演算子であり、以下の 3 種類がある。括弧内はクエリ中の演算子を示している。

- 水平結合子 (,)

データを横に結合して出力。

例: Name, Tel

name	tel
------	-----

- 垂直結合子 (!)

データを縦に結合して出力。

例: Name! Tel

name
tel

- 深度結合子 (%)

データを 3 次元方法へ結合。出力が HTML ならばリンクとなる。

例: Name % Tel

name

 →

tel

2.2 反復子

反復子は指定する方向に、データベースの値があるだけ繰り返して表示する。また反復子はただ構造を指定するだけでなく、そのネストの関係によって属性間の関連を指定できる。例えば

[科目名]!, [学籍番号]!, [評点]!

とした場合には各属性間に関連はなく、単に各々の一覧が表示

されるだけである。一方、ネストを利用して

[科目名! [学籍番号, 評点]!]!

とした場合には、その科目毎に学籍番号と評点の一覧が表示されるといったように、属性間の関連が指定される。以下、その種類について述べる。

- 水平反復子 ([],)

データインスタンスがある限り、その属性のデータを横に繰り返し表示する。

例: [Name],

name1	name2	...	name10
-------	-------	-----	--------

- 垂直反復子 ([]!)

データインスタンスがある限り、その属性のデータを縦に繰り返し表示する。

例: [Name]!

name1
name2
...
name10

2.3 装飾子

SuperSQL では関係データベースより抽出された情報に、文字サイズ、文字スタイル、横幅、文字色、背景、高さ、位置などの情報を付加できる。これらは装飾演算子 (@) によって指定する。

<属性名>@{ <装飾指定> }

装飾指定は”装飾子の名称 = その内容”として指定する。複数指定するときは各々を,” で区切る。

2.4 関数

SuperSQL ではいくつかの関数が用意されている。ここでは代表的な関数を 4 つ紹介する。

2.4.1 imagefile 関数

imagefile 関数を用いると画像を表示することが可能となる。引数には属性名、画像ファイルの存在するディレクトリにパスを指定する。

imagefile(id, path=”./pic”)

2.4.2 link 関数 (出力メディアが HTML の場合のみ)

link 関数は FOREACH 句と同時に用いる。これらを用いることで深度結合と同様にリンクを生成することができる。

link(cou.name, file=”./menu.sql”, att=co.country)

2.4.3 invoke 関数

invoke 関数はリンクを生成するための関数である。link 関数の場合、SuperSQL を手動で実行することでリンク先を生成しておくが、invoke 関数の場合、ユーザのリクエストに応じて動的にリンク先を生成する。

invoke(cou.name, file=”./menu.sql”,
condition=”ca.country=”+co.country)

2.4.4 embed 関数

embed 関数を用いることでクエリを分割・合成することが可能になる。利用方法は別ファイルに保存されたクエリ、もしくは HTML ファイルを埋め込みたい箇所に embed 関数を記述す

る。

embed(file=”./test.sql” where=”ca.id=” att=ca.id)

3. SuperSQL での HTML5 の対応

HTML から HTML5 に進化した上で SuperSQL で対応した機能を以下に示す。

3.1 Canvas の使用

Canvas とは、ブラウザ上に図を描くために策定された仕様であるが、本研究では数値データを視覚的に表現出来る meter 関数の使用において Canvas を使用している。

3.2 新たなタグへの対応

html5 になる事で新しく追加されたタグ (video, audio, details, ruby) などがあるが、各々関数を定義する事により、HTML ページに対して新たに出力している。

3.3 新たなフォーム機能

HTML5 のフォーム機能は HTML と比較し、url, mail 入力用の専用フォームの導入、バリデーション機能が追加された。従来の SuperSQL でも form の作成は出来たが、HTML5 の form 機能を細かく設定することは出来なかった。本研究では装飾子で type を指定する事により、その type に合った専用フォームとバリデーション機能の提供を自動で行えるようにして従来の SuperSQL と比べ form 要素の出力を多彩にした。

3.4 オフライン Web アプリケーション機能

HTML5 の新機能で CacheManifest というものがある。それは manifest ファイルにどのファイルをキャッシュするかを記述しておくことにより、初回アクセス時に指定されたファイルをキャッシュすることをブラウザに指示し、次回はページにアクセスする事無く、ローカルからファイルを見ることができるようになる。それによって起動の高速化、Web ページのオフライン WEB アプリ化などができる。

その際の manifest ファイルの記述は以下のように記述する必要がある。

```
— ファイル名.manifest —
CACHE MANIFEST
CACHE:
キャッシュしたいファイル名のパス
...
```

また、キャッシュされる HTML ファイルのヘッダーに< html manifest=“ ファイル名.manifest ”> と、どんなファイル名の manifest ファイルが存在する事を記述する必要がある。

SuperSQL の実行時に引数として-cache を記述する事により、自動で Manifest ファイルを生成する機能を実装した。

4. 提案システム

SuperSQL に加えた関数や機能のシステムに関して述べる。

4.1 meter 関数

meter 関数を用いると数値データを canvas を使用したメー

ター画像として表示する事が可能となる。引数には数値データを持つ属性名を指定する。装飾子で以下の値を指定する事でメーターを装飾する事ができる。なお、指定しない場合は初期値が入る。

図1は数値データの表とそれを meter 関数で装飾したものの比較図である。装飾子指定で様々なメーターを表示する事ができる。

- width, height... メーターを囲うテーブルの大きさ
- m_width, m_height... メーターの大きさ
- max, min... メーターの最小, 最大値
- low, high... メーターの閾値
- max_color, low_color, high_color, mid_color, bg_color... メーターの色

記述例: meter(id)@{width = 100, height= 30}

8	50	85	85	8			
9	60	110	110	9			
4	45	60	60	4			
6	80	120	120	6			
10	35	55	55	10			
3	120	85	85	3			
7	40	70	70	7			
11	70	90	90	11			
5	55	85	85	5			
1	70	45	45	1			
2	95	65	65	2			

図1 数値表とメーター実行例

4.2 movie 関数

movie 関数を用いると html5 の video タグを利用した動画プレーヤーと再生する動画を表示する事ができる。引数には属性名と動画ファイルの存在するディレクトリのパスを指定する。
記述例: movie(id, path= "/movie")

4.3 Form 要素の type 指定

SuperSQL では更新ページや新規追加ページを作る時に id@insert などの表記をするとフォームを作る事ができるが、装飾子で type を指定する事により type に合ったフォームの型とそれに合ったバリデーション機能をフォームに与える事ができる。type で指定できるものは (mail, age, tel, url) の4種類であるが、type=" \d2,4-\d2,4-\d4 "のように直接正規表現を type に直接指定してもバリデーション機能を扱う事ができる。図2では記述部分の type を mail にしているのに、メールアドレスの文法に合わない記述をした際に示されるエラーメッセージを示している。

id@{type="mail"}

1	ishizaki	test@test.jp	22	0000-2020-2020	http://aaa.aaa.jp
2	aoyama	cha@matui.matui	23	090-0901-0909	http://aaa.aaa.jp2
4	zonop		43	090-2222-2222	
		aaa			

図2 type 指定をした form の実行例

4.4 ruby 関数

ruby 関数を用いると文字列にふりがなをふる事が可能となる。引数には主として表示する属性名とふりがなとして使用する属性名を指定する。図3のように、非対応ブラウザでは括弧内になを出力する。

記述例: ruby(id, yomi=id2)

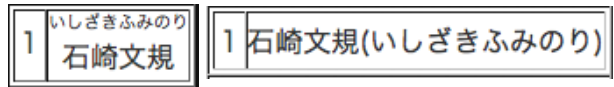


図3 ruby 実行例 (左:対応ブラウザ, 右:非対応ブラウザ)

4.5 detail 関数

detail 関数を用いるとフローコンテンツのあるページ作成が可能となる。引数には主として表示する属性名とフローコンテンツに使用する属性名 (i1, i2, ..., in) を指定する。装飾子として (f_1, f_2, ..., f_n) と (e_1, e_2, ..., e_n) を指定する事により、n 番目のコンテンツの前後に固定文字列を挿入できる。図4の ishizaki をクリックする事より、図のように詳細が開く動作となっている。

記述例: detail(id, i1=id1, i2=t.id2)@{f_i1=hoge;, e_i1=hoge2;}

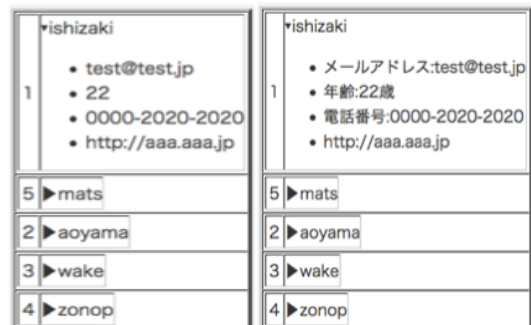


図4 detail 実行例 (右が装飾子有)

4.6 cachemanifest 自動生成

SuperSQL 実行時の引数として -cache と指定 (記述例: ssl q1.sql -cache) する事により、HTML5 のオフライン WEB アプリケーション作成に必要な manifest ファイルを自動生成する。SuperSQL では % や link 関数を使ったリンク先ページの生成があるが、manifest ファイルは SuperSQL クエリを実行した回数毎に出力され、ファイル名はそのクエリで作られるファイル名.html のファイル名を使ったファイル名.manifest となる。これによって、manifest ファイルは link 関数を使用して複数回クエリを実行時も別なファイル名の manifest ファイルが出力されるため、上書きされることは無い。また、imagefile 関数や video 関数で利用する画像や動画もキャッシュの対象とする場合

imagefile(name, path= "/pic")@{cache}

のように関数の装飾子に cache と記述しておく事により、読み込まれる画像や動画もキャッシュの対象として manifest ファイルに追加される。動作例として

q1.sql

```
GENERATE HTML5 [  
link(t7.name,file="q2.sql",att=t7.id)  
]!  
FROM test7 t7
```

q2.sql

```
FOREACH t7.id  
GENERATE HTML5 [  
t2.id,t2.name,t7.name,  
imagefile(t2.pic,path=../pict)@{cache}  
]!  
FROM test7 t7,test2 t2  
WHERE t7.id=t2.id
```

のように二つのクエリを実行したとする。この時に生成されるファイルは q1.sql では [q1.html] , q2.sql では [q2.1.html, q2.2.html, q2.3.html, q2.4.html] となり, q2.sql によって表示される画像が [test1.jpg, test2.jpg, test3.jpg, test4.jpg] だったとする。この時, manifest ファイルは q1.manifest と q2.1.manifest の 2 つが生成され, ファイルの中身は以下のようになる。

q1.manifest

```
CACHE MANIFEST  
CACHE:  
q1.html
```

q2.1.manifest

```
CACHE MANIFEST  
CACHE:  
q2.1.html  
q2.2.html  
q2.3.html  
q2.4.html  
../pict/test1.jpg  
../pict/test2.jpg  
../pict/test3.jpg  
../pict/test4.jpg
```

また, 生成された q1.html ファイルには<html manifest="q1.manifest">, それ以外には<html manifest="q2_1.manifest"> の記述がヘッダ部分に追加される。

5. 評価

本研究の HTML 生成の有用性を検討するために情報工学科の学生 15 名に対しアンケートを行った。被験者は上記の 3 通りの方法で好きなように提示されたデータを表示する HTML

ページを作成してもらった。その後 5 項目について、そう思う or できた・どちらかと言えばそう思う or どちらかと言えばできた・どちらとも言えない・どちらかと言えばそう思わない or どちらかと言えばできていない・そう思わない or できていない、の 5 段階で答えてもらい、質問の内容は以下に示す。また、回答理由についても答えてもらう。

Q1. 従来の SuperSQL と比較して数値データが有効利用できたか？

Q2. タグ打ちと比較して作業が簡単であったか？

Q3. 作りたい機能をクエリですぐに実装することができたか？

Q4. 従来の SuperSQL と比較して表現力が向上したように感じたか？

Q5. タグの手打ちのものと比較して満足する表現ができたか？

5.1 数値データの有効利用

図 5 を見ると被験者の 80%以上が従来の SuperSQL と比較して数値データを有効利用できたと回答している。回答理由には meter 関数を利用する事によって、数値データを視覚的にわかりやすく表現する事ができた。と言う意見が多数を占めた。有効利用出来なかったと回答している回答理由の中には、数値データがメーターを使って表現するのに向いていないものもあると言ったものが挙げられた。回答理由から、似た種類の数値データを一度に表現する場合は meter 関数を利用する事によって数値データを有効利用出来ると考えられる。逆に、身長や体重と言ったデータを表現するときは meter 関数を使うべきでは無いので、従来の SuperSQL と比較して数値データの有効利用はほとんど変わらないと考えられる。

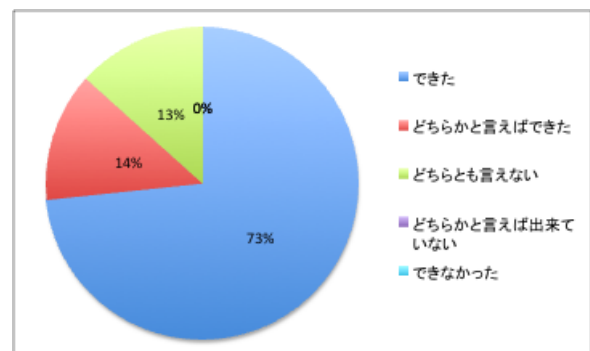


図 5 従来の SuperSQL と比較して数値データが有効利用できたか？

5.2 SuperSQL クエリ記述の検討

この節では、SuperSQL クエリを記述する上でタグの手打ちとの比較、表現したい HTML をクエリですぐに実現出来るかと言った事を検討する。

図 6 において被験者の 70%以上がタグの手打ちと比較して作業が簡単だったと回答している。回答理由には、純粋にコピー & ペーストを利用してタグを手打ちするよりも、本研究の方がコードの記述量が圧倒的に少ない、タグを書くよりも関数を使った方が簡単に実装ができたと言った意見が挙げられた。逆に、作業が簡単でなかった理由としては、HTML のタグを使

う事に慣れていて SuperSQL を使った生成に慣れていないので慣れ親しむのに時間がかかったという意見が挙げられた。

図 7 においては作りたい機能をすぐに実装出来たという意見が約 40%，すぐに実装出来なかったという意見が 33%であり，大きな差は見られなかった。すぐに実装出来なかった回答理由としては，細かく設定する時に装飾子指定が多すぎるので装飾子の指定に時間がかかったという意見が複数挙げられていた。

回答理由から，タグの手打ちよりはコード記述の量は大幅に軽減する事ができるが，細かい装飾を行う場合，実現するには作業量が増えることが考えられる。

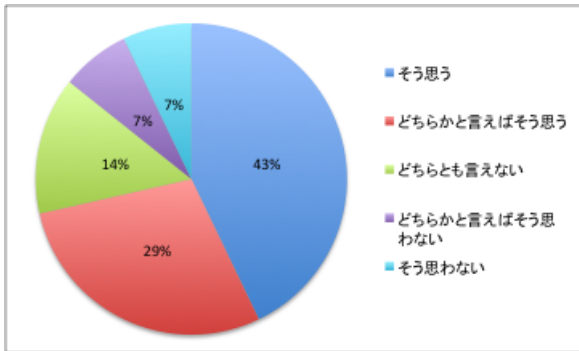


図 6 タグ打ちと比較して作業が簡単であったか？

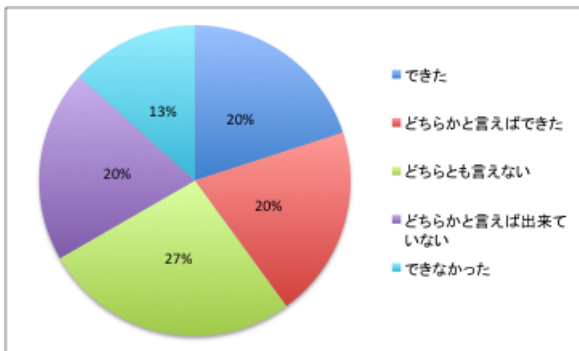


図 7 作りたい機能をクエリですぐに実装することができたか？

5.3 既存手法との表現比較

この節では，従来の手法，タグの手打ちとの表現力の比較について考察を行う。

図 8 において被験者の 80%以上が従来の SuperSQL よりも表現力が向上したと答えている。回答理由には，本研究の SuperSQL は従来の SuperSQL の機能が使えるので，本研究で実装された分だけ表現力が向上していると言った意見が多数であった。

図 9 においてはタグの手打ちと比較して表現力が向上したという意見が約 40%，向上していないという意見が 47%であった。表現力が向上した回答理由としては，Javascript などに慣れ親しんでいないので Canvas を利用する事ができないが，本研究なら簡単に Canvas を使った出力ができると言ったものが挙げ

られ，表現力が向上していないといった回答理由は，全ての新たな HTML5 のタグにまだ対応していないため，Canvas を使いこなせる人にとっては表現が限定的すぎる。と言った意見が挙げられた。

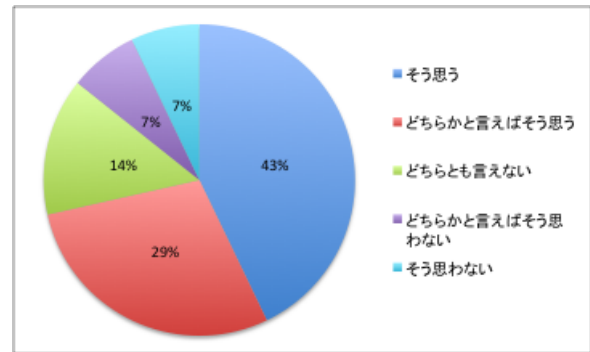


図 8 従来の SuperSQL と比較して表現力が向上したように感じたか？

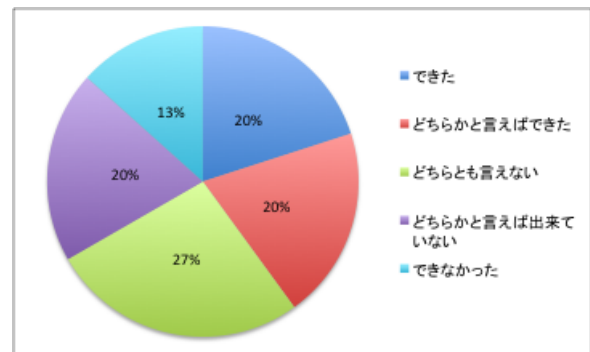


図 9 タグの手打ちのものと比較して満足する表現ができたか？

6. ま と め

本研究では従来の SuperSQL の HTML 生成において，まだ対応していなかった HTML5 の新機能の対応を実現するために新たな関数の導入や，manifest ファイルの自動生成機能を実装することにより SuperSQL で HTML5 の機能を持ったページの出力を実現をした。SuperSQL はデータベース中の情報を用いて少ない記述量でほとんどの質の Web ページ作成を行えるという特性があるが，本研究でもこの特性は変わらない。今後は装飾子指定により，より細かいレイアウトを行えるような改良，未対応タグへの対応，未対応ブラウザで開いた時の表示方法の改良をやっていきたい。

文 献

- [1] SuperSQL: <http://SuperSQL.db.ics.keio.ac.jp/>
- [2] M. Toyama: "SuperSQL: An Extended SQL for Database Publishing and Presentation", Proceedings of ACM SIGMOD '98 International Conference on Management of Data, pp. 584-586, 1998
- [3] 有園チエ，遠山元道: 『WEB アプリケーション開発のための SuperSQL の拡張』，慶應義塾大学開放環境科学専攻修士論文，2009 年度
- [4] HTML5.jp: <http://www.html5.jp/>