

SuperSQL を用いたデータベース中心の Web ページ生成の実現

春野 健吾[†] 五嶋 研人[†] 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: [†]{halken,goto}@db.ics.keio.ac.jp, ^{††}toyama@ics.keio.ac.jp

あらまし 情報技術の発達により、インターネットを利用する機会が増え、情報発信のために様々な人が Web ページを生成する機会も増えてきた。しかし、モダンな Web ページを生成するためには HTML や CSS, JavaScript など、様々な言語の知識が必要であり、レイアウト構造を表現するだけでも複雑な記述が必要となる。そこで、短いコード量で複雑なレイアウト表現を可能とする言語として、SuperSQL がある。SuperSQL とは、SQL の拡張言語であり、関係データベースの出力結果を構造化し、多彩なレイアウト表現を可能にするものである。そこで本研究では SuperSQL を用いたデータベース中心の Web ページ生成機構の実装を提案する。SuperSQL のクエリ文のみを記述することによって、Web ページ全体のレイアウト構造を持つ HTML や装飾を持つ CSS を生成する。これにより関係データベースの出力結果を Web コンテンツとして利用する SuperSQL の特性を活かしながら、複雑でモダンな Web ページを短いクエリ文で生成することができる。

キーワード SuperSQL, SQL, HTML, CSS, JavaScript, Web

1. はじめに

現在の情報技術の発達により、多くの人がインターネットを利用する機会が増えてきた。それに伴い、ニュースや宣伝、ブログなどによる情報発信のために様々な人が Web ページを生成するようになった。しかしながら、Web ページを生成するためにはマークアップ言語である HTML の知識が必要であり、さらに多彩なデザインを施したり、動的なページを生成する場合には加えて CSS や JavaScript などの知識が必要となる。そのため、一から Web ページを生成するには多くの時間を要し、開発者の負担が大きい。

そこで、短いコード量で複雑なレイアウト表現を可能とする言語として、SuperSQL [1] [2] がある。SuperSQL とは、SQL の拡張言語であり、関係データベースの出力結果を構造化し、多彩なレイアウト表現を可能にするものである。これを用いて、関係データベースの出力結果を HTML ファイルとして生成することが現在可能であるが、テーブルとして表現するという体裁を守っているため、広く一般的に公開されているような Web ページの生成には適しておらず、SuperSQL における HTML 生成機構の価値を損ねている。本研究では、従来の SuperSQL の HTML 生成機構を大幅に改変し、SuperSQL を用いたデータベース中心の Web ページ生成機構を提案する。Web ページ開発者が、SuperSQL のクエリ文のみを記述することによって、Web ページ全体のレイアウト構造を持つ HTML や、装飾デザインの情報を持つ CSS を生成する。また、装飾演算子や関数を用いて記述することで JavaScript で実装されるような動的なページを生成する。関係データベースの出力結果を Web コンテンツとして利用する SuperSQL の特性を活かしながら、複雑でモダンな Web ページを短いクエリ文で生成することができる。

2. 関連研究

Web ページを生成するにあたり、HTML や CSS を少なくとも用いる必要がある。これらの言語を組み合わせることで自由自在なレイアウトデザイン表現を行うことができる。近年では、Web ページ開発者がより簡単にプログラミングできるような研究がいくつかなされている。HTML と CSS は他のプログラミング言語とは違ってエラーを表示しないため、何か記述ミスによって予期せぬレイアウト崩れが起こった場合に、修正するには困難である [3]。openHTML [4] は、HTML におけるエラーと CSS におけるエラーを区別して表示するため、修正が容易に行え開発者の負担を少なくする。WEBNAT [5] は、Web アプリケーション生成におけるクライアント・サーバプログラミングの負担を減らすべく開発されたフレームワークであり、記述ルールを抽象化することで送信フォーマットやデータの動的表示などを容易に行う。

Web コンテンツを HTML に直接記述して持たせるのではなく、別の媒体と結びつけて Web アプリケーションを生成する研究がなされている。Quilt [6] では、情報管理や計算処理を行うスプレッドシートを HTML と結びつけることによって、動的な Web アプリケーションを簡単に作成することができる。スプレッドシートは主に情報管理として使用されるので、情報管理システムとしての Web アプリケーション作成において非常に有益であると考えられる。しかしながら、関係データベースに比べてシンプルな情報管理となるため、より複雑で多彩な Web ページを実現するには難しいと考えられる。また、近年 WordPress [9] や Joolma [10]、Drupal [11] などの Web コンテンツ管理システム (CMS) が広まりつつある [7] [8]。CMS はあらかじめ好みの HTML テンプレートを用意し、ブラウザ上でコンテンツを GUI で追加して Web ページを生成するシステムである。直感的に操作でき、わかりやすいため、Web ページ開発

初心者にとっては非常に便利である。しかしながら、開発者好みの HTML テンプレートを一から作る場合には HTML や PHP の知識が必要であるため、デザインにおけるカスタマイズ性に乏しいと考えられる。

3. SuperSQL

SuperSQL [1] [2] 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 の拡張言語であり、慶應義塾大学遠山研究室で開発されている。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 や XML, 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。

3.1 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する。

例: name, place

name	place
------	-------

- 垂直結合子 (!)
データを縦に結合して出力する。

例: name! place

name
place

- 深度結合子 (%)
データを 3 次元方向へ結合する。出力が HTML ならばリンクとなる。

例: name % place

name	→	place
------	---	-------

3.2 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はなく、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

[部署名]!, [雇用者名]!, [給料]!

とした場合には各属性間に関係はなく、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

[部署名! [雇用者名, 給料]!]!

とした場合には、その部署毎に雇用者名と給料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て表示する。

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て表示する。

例: [Name]!

name1
name2
...
name10

3.3 複合反復子

複合反復子は反復子を組み合わせたものである。定義は以下のとおりである。

[<TFE>]<結合子><回数><結合子>

ある方向での反復の上限を指定し、これを超える場合に次に指定した方向に伸長することを指定する。例えば下記の場合は、要素を 3 つ横結合した後、縦結合する。

例: [Name],3!

name1	name2	name3
name4	name5	...

3.4 装飾演算子

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。

[name@{width=“100”, color=“red”}]!

3.5 関数

SuperSQL では、出力の表現を高め、様々なデータ型を取り扱うため、様々な関数が用意されている。関数の種類に関して、4.2.3 章で示す。

3.6 HTML 生成機構の問題点

SuperSQL では現在、関係データベースの出力結果を HTML ファイルとして生成することができ、短いコード量で多彩な HTML 形式のテーブルレイアウト表現を可能にする。しかし、table タグを用いたテーブルとして表現するという体裁を守っているために、装飾表現が大きく制限され、広く一般的に公開されるような Web ページとして生成するには適していない。そこで本研究では、SuperSQL の内部に新たに Web ページ生成機構を実装し、関係データベースの情報を Web コンテンツとして活用することで、様々なレイアウトデザインを表現可能にし、少ないコード量で Web ページを生成する。

4. Web ページ生成機構

4.1 内部設計

SuperSQL の処理系は、構文解析部、リスト構造生成部、コード生成部から構成されている。図 1 に SuperSQL の内部処理を示す。SuperSQL クエリを読み込み、構文解析部で構文解析を行い、SQL クエリ文とレイアウト式を生成する。その SQL クエリ文で関係データベースに問い合わせ、出力結果をリスト構造生成部へ送る。リスト構造生成部で出力結果を階層型リスト構造に変換し、コード生成部で階層型リスト構造とレイアウト式を組み合わせ、指定した出力媒体を生成する。今回は、構文解析部における構文解析ルールを追加し、さらにコード生成部に新たに Web ページ生成機構を追加する。

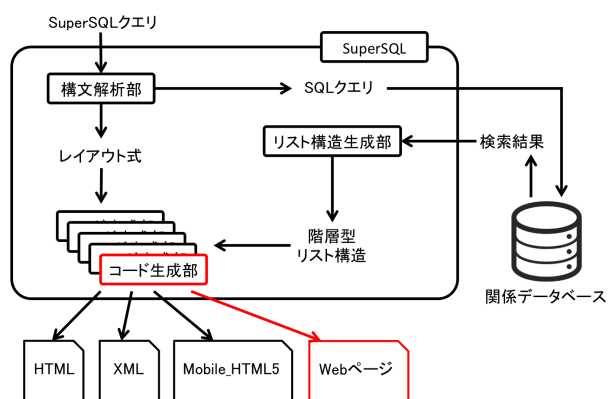


図 1 SuperSQL の内部処理

4.2 レイアウト表現

SuperSQL では、出力結果を構造化し、Web ページ全体のレイアウトを形成する他、一部に対してテーブル表現やリスト表現、などを形成する。また、装飾演算子に関係データベースの要素を指定することで、各タプルごとに具体的な装飾を指定することができる。

4.2.1 レイアウト

結合子 (3.1) を用いて Web ページ全体のレイアウトを生成する。この時、装飾演算子による幅の指定がない場合、全体幅をシステム既定値 (1280px) となるようにレイアウトする。また、反復子 (3.2) や複合反復子 (3.3) を用いることで関係データベースの出力結果を Web コンテンツとして利用し、多彩なレイアウト表現ができる。従来は table タグによるレイアウト表現を行っていたが、Web ページ生成機構では div タグによってレイアウト表現を行っている。例として、図 2 にレイアウトに関するサンプルクエリとそれに基づき生成されたレイアウトを示す。図 2 のように、サブメニューや記事コンテンツのレイアウトを反復子を用いて表現でき、追加したい場合はクエリを変えることなく関係データベース側で情報を追加することで実現する。デフォルトではレイアウトの枠線が表示されないようになっているが、装飾子を用いて @{border="on"} と指定することでうすい枠線を表示する。

Web ページ生成機構では、全体のレイアウトを div タグを用

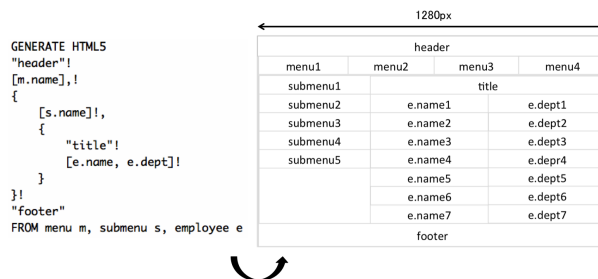


図 2 SuperSQL サンプルクエリとレイアウト表現

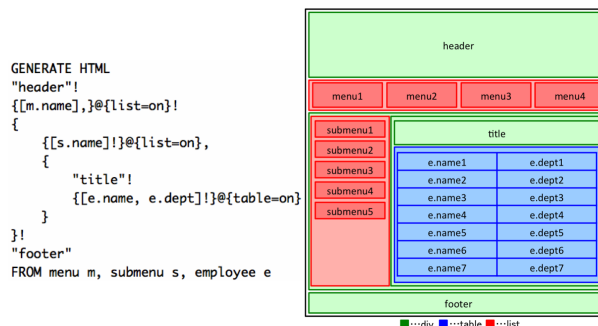


図 3 装飾演算子による部分的なレイアウト表現

いて生成しているが、部分的に装飾演算子で指定することによって、図 3 のように従来の table タグによるレイアウト表現など、div タグ以外によるレイアウト表現が可能となる。以下、その種類について述べる。

- テーブル表現 @{table="on"}

table タグを用いたテーブルレイアウト表現をする。

- リスト表現 @{list="on"/@"number">}

ul タグや li タグを用いたリスト表現をする。@{list="on"} と指定すると箇条書きによるリスト、@{list="number"} と指定すると順序付きリストとなる。

4.2.2 装飾子によるデザイン表現

装飾演算子 (3.4) を用いることで多彩なデザイン表現を行うことができるが、装飾指定の内容を関係データベース要素で指定することができる。図 4 に関係データベースを利用した装飾演算子の指定方法を示す。テーブル内に装飾内容を格納する属性を追加する。その属性名を装飾演算子の装飾内容に記述することで各タプルごとの装飾を施すことができる。

```
GENERATE HTML
[t.subject, [t.name@{bgcolor=t.bgcolor}]]! ]!
FROM teacher t
```

teacher table

id	name	subject	bgcolor
1	佐藤	国語	yellow
2	齋藤	数学	blue
3	清水	国語	yellow
4	渡辺	英語	red
5	菊池	数学	sky
6	高橋	国語	yellow
7	金子	英語	pink

国語	佐藤
	清水
	高橋
数学	齋藤
	菊池
英語	渡辺
	金子

図 4 関係データベースを利用した装飾内容指定

```

GENERATE HTML
[{{t.subject, [t.name]!}}@{{bgcolor=t.bgcolor}}]!
FROM teacher t

```

teacher table

id	name	subject	bgcolor
1	佐藤	国語	yellow
2	齋藤	数学	blue
3	清水	国語	yellow
4	渡辺	英語	red
5	菊池	数学	sky
6	高橋	国語	yellow
7	金子	英語	pink

国語	佐藤
	清水
	高橋
数学	齋藤
英語	菊池
	渡辺
	金子

図 5 関係データベースを利用した複数装飾内容指定

また、図 5 のように複数の要素に指定することもできる。しかし、条件としてグルーピングされたタプルで装飾内容が共通の場合のみ適用される。図 5 を例とすると、国語に関しては全て bgcolor 要素が yellow なので反映されるが、数学と英語は bgcolor 要素がそれぞれ異なっているため反映されない。

4.2.3 関数によるデザイン表現

SuperSQL では、Web ページ生成における出力の表現のために様々な関数が用意されている。ここでは、代表的な関数をいくつか紹介する。

• image 関数

image(< 画像ファイル名 >, < ディレクトリのパス >)

image 関数を用いることによって、画像を表示することができる。画像ファイル名の場所に、関係データベースの属性名を参照することも可能であるので、あるディレクトリに画像ファイルを用意し、各タプルに pict という属性を追加し画像ファイル名を格納することで、反復子を用いれば一度に画像を表示することができる。下記のクエリは、画像を 3 列に画像を表示する記述例である。

```
[image(pict, “./image”),],3!
```

• anchor 関数

anchor(< リンク名 >, < アドレス名 >)

anchor 関数を用いることによって、リンクを生成することができる。リンク名、アドレス名ともに関係データベースの属性名を参照することも可能であるので、リンクに関するテーブルを作成しておけば、一行で複数のリンクを生成できる。下記のクエリは、m.name にリンク名、m.link にアドレス名を格納する menu テーブルから参照してメニューを生成する記述例である。

```
[anchor(m.name, m.link)],
```

• header 関数

header(< ヘッダー名 >)

header 関数を用いることによって、ヘッダーを生成することができる。ページをスクロールしても、必ず上部にレイアウトされる。

• footer 関数

footer(< フッター名 >)

footer 関数を用いることによって、フッターを生成することができる。ページをスクロールしても、必ず下部にレイアウトされる。

4.3 ダイナミック表現

Web ページ生成において、CSS を用いた見栄えのデザインの指定の他に、JavaScript などを用いたダイナミックな表現がある。SuperSQL では関係データベースと組み合わせてスライドショーやアコーディオンメニューなど、ダイナミックな表現を行うことができる。

4.3.1 スライドショー

装飾演算子を用いて@{slide=“on”} と指定することでスライドショーの表現ができる。image 関数と反復子を用いて実装する。水平反復子ならば横に、垂直反復子ならば縦にスクロールする。

```
{{image(pict, “./image”)!}}@{slide=“on”}
```

4.3.2 アコーディオン

装飾演算子を用いて@{accordion=“on”} と指定することでアコーディオンの表現ができる。深度結合子と合わせて実装する。本来ならば深度結合子(%)を用いると自動的にリンクを生成するが、アコーディオンが適用されると、リンクを生成せずアコーディオンに隠れる部分を形成する。例えば下記のクエリでは、menu の要素をクリックすると submenu の要素が出現するようになる。

```
{{menu% [submenu]! }}@{accordion=“on”}
```

4.3.3 ツールチップ

装飾演算子を用いて@{tooltip=“on”} と指定することでツールチップの表現ができる。アコーディオンと同様に深度結合子と合わせて実装する。例えば下記のクエリでは、content の要素をマウスオーバーすると option の要素が出現するようになる。

```
{{content% option! }}@{tooltip=“on”}
```

4.4 レスポンシブ表現

近年、スマートフォンやタブレット端末の普及により、モバイル端末を用いて Web サイトを閲覧するように機会が多くなっている。そのため、ウィンドウサイズの大きさに合わせてページのデザインを最適化するため、レスポンシブな Web ページを生成する必要がある。SuperSQL では、装飾演算子を用いて@{responsive=“on”} と指定することにより、生成する Web ページをレスポンシブ対応するようにしている。また、レスポンシブ表現における具体的な指定も装飾演算子を用いることで可能である。

レスポンシブ指定した場合、ウィンドウサイズに対応して自動的にレイアウトを変更する。この時、SuperSQL クエリにお

ける結合子, 反復子のレスポンス対応におけるルールに基づいてレイアウトを変更する. そのルールの定義について, 以下に示す.

(1) ウィンドウサイズの幅がある水平反復子で形成される幅よりも小さい場合, その水平反復子を複合反復子に変換する.

(2) ウィンドウサイズの幅がある水平結合子で形成される幅よりも小さい場合, その水平反復子内の一番階層の深い水平結合子を垂直結合子に変換する.

例として, 以下のようなサンプルクエリがある.

```
header! [menu],! {[submenu]!,
    {title! {[news]!, [content]!}}! footer
```

このクエリからなるレイアウトと階層の木構造は図 6 のようになる. 次に, ウィンドウサイズが小さくなり, menu における水平反復子の幅よりも小さくなったとする. その時, 上記の定義より, 水平反復子を複合反復子に変換する. 折り返す数は, 元のサイズで横結合していた要素数の半分とする. そのようにして変換されたものを図 7 に示す. さらにウィンドウサイズが小さくなり, 今後は submenu などを含む要素を水平結合子で形成する幅よりも小さくなったとする. その時, 上記の定義より, その水平反復子内の一番階層の深い水平結合子を垂直結合子に変換する. 今回の場合, news と content を結合する水平反復子が一番階層の深い水平結合子となっているため, この結合子を垂

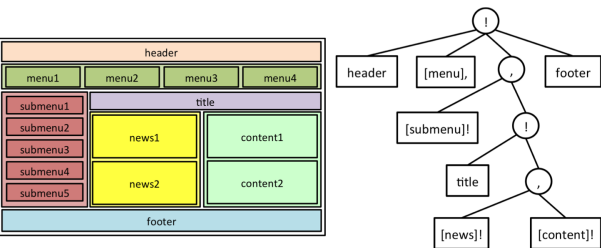


図 6 サンプルクエリのレイアウトと木構造

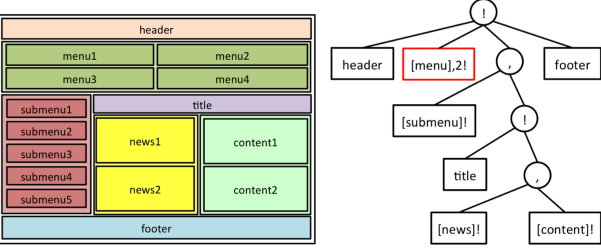


図 7 水平反復子から複合反復子への変換

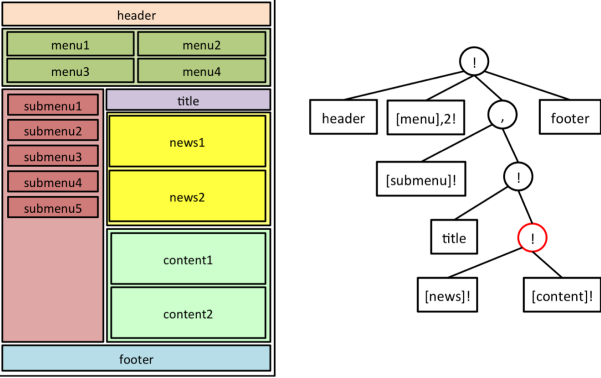


図 8 水平結合子から垂直結合子への変換

直結合子に変換する. 変換されたものを図 8 に示す. このようにしてウィンドウサイズに合わせて結合子や反復子を適時変更することでレスポンスに対応する.

5. 評価

図 9 のようなネット通販サイトを想定したサンプル Web ページを, HTML と CSS を用いて生成する方法, PHP を用いて生成する方法, そして SuperSQL を用いて生成する方法の 3 種類で同じ Web ページを生成し, Web ページ生成に必要とするコード量の比較を行った. なお, 動的な表現に関しては jQuery を用いている.

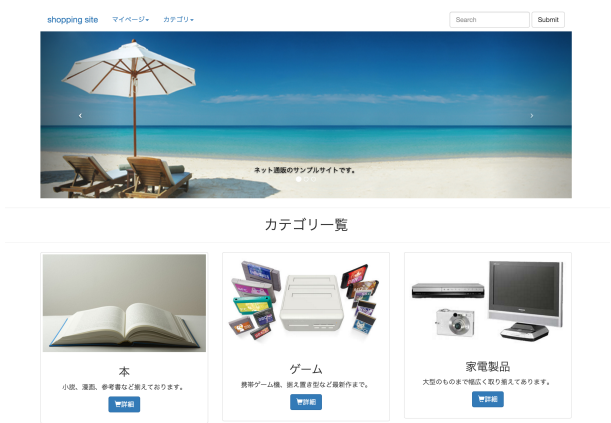


図 9 評価に用いられたサンプル Web ページ

比較を行った結果, 表 1 のようになった. HTML と CSS を用いた場合と PHP を用いた場合と比べて, 非常に少ない行数で同じ Web ページを再現できることがわかる. 行数を少なく抑えられた要因として 3 つ考えられる. 1 つ目は, Web ページのレイアウト構造を従来ならば DOM 構造で階層的に記述する必要があったが, SuperSQL では結合子を用いてレイアウト表現を行っているため, 階層表現を最小限にしていることである. 2 つ目は, CSS で表現する装飾を SuperSQL ではデータベースに保管し, クエリの属性名を用いて代替的に装飾表現を行っているため, コードとして書き起こす量を抑えていることである. 3 つ目は, レスポンスな Web ページを表現するために従来では多くの記述を要するが, SuperSQL では自動的にレスポンス表現をするためそれに関する記述を必要としていないことである. これら 3 つの要因によって, コード量を減らすことができ簡潔になるため, Web ページの変更や修正を容易にできるようになることが期待される. 今回は非常に簡単な Web ページを生成して評価したが, 今後はサイト規模でページ生成する場合どれくらいのコード量の差が生じるのか, また Web ページ開発者に SuperSQL を使用してもらい, 作業量における評価も検討している.

表 1 コード量の比較		
HTML + CSS	PHP	SuperSQL
652 行	421 行	32 行

6. おわりに

6.1 結 論

本研究では, SuperSQL に新たに Web ページ生成機構を実装することにより, データベース中心の Web ページ生成を実現する。HTML や CSS など, Web ページ制作に必要な多くの言語知識なしに, SuperSQL のみを用いて Web ページを生成できる。また, レイアウト表現やダイナミック表現, レスポンシブ表現など様々な表現を, 一般的な言語でコーディングするよりも少ないコード量で実現することができる。

6.2 今後の課題

今回提案している Web ページ生成機構では, HTML と CSS, JavaScript を SuperSQL クエリから生成する。そのため, PHP を用いて実装するようなフォームの送信や検索などの機能が現段階では実装できない。今後は, 従来 PHP で実装できる一般的な機能を SuperSQL のみで生成できるようにしていくほか, Web ページで頻繁に用いられるレイアウトや装飾, アニメーション等を適時追加実装していきたいと考えている。

また, SuperSQL を用いるためには, 少なくとも SQL などの関係データベース言語の知識が必要となる。よって現在の SuperSQL はプログラミング熟練者向けのクエリ言語となっていて, 手軽に Web ページを生成したいプログラミング初心者でも容易に扱えるようにすべきである。この問題を解決するために, SuperSQL を Web コンテンツ管理システムに導入するシステムを実装し, 関係データベースに関する情報管理を GUI で, レイアウトデザインの生成を SuperSQL で分割して行うことで Web ページを生成する手法ができないか考察し, 研究をすすめていく。

文 献

- [1] SuperSQL: <http://ssql.db.ics.keio.ac.jp>
- [2] Motomichi 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] Park, Thomas H., et al, “Towards a taxonomy of errors in HTML and CSS,” Proceedings of the ninth annual international ACM conference on International computing education research, pp.75-82, 2013.
- [4] Park, Thomas H., Brian Dorn, and Andrea Forte, “An Analysis of HTML and CSS Syntax Errors in a Web Development Course,” ACM Transactions on Computing Education (TOCE), 2015.
- [5] Sinha, Nishant, Rezwana Karim, and Monika Gupta, “Simplifying web programming,” Proceedings of the 8th India Software Engineering Conference, pp.80-89, 2015.
- [6] Benson, Edward, Amy X. Zhang, and David R. Karger, “Spreadsheet driven web applications,” Proceedings of the 27th annual ACM symposium on User interface software and technology, pp.97-106, 2014.
- [7] Souer, Jurriaan, et al, “Situational requirements engineering for the development of content management system-based web applications,” International Journal of Web Engineering and Technology, pp.420-440, 2007.
- [8] Souer, Jurriaan, et al, “Engineering a design method for web content management implementations,” Proceedings of the

10th International Conference on Information Integration and Web-based Applications & Services, pp.351-358, 2008.

- [9] WordPress: <https://wordpress.org>
- [10] Joomla: <https://www.joomla.org>
- [11] Drupal: <https://www.drupal.org>