

SuperSQL を用いたモダンな Web ページの生成

大山 直輝[†] 五嶋 研人[†] 遠山 元道^{††}

[†] 慶應義塾大学大学院理工学研究科 〒223-8522 神奈川県横浜市港北区日吉 3-14-1

^{††} 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: [†]{yama,goto}@db.ics.keio.ac.jp, ^{††}toyama@ics.keio.ac.jp

あらまし 近年、モダン Web という概念が広がりを見せている。これは、1 つの Web ページ内に複数のコンテンツを含み、ページ遷移なしにスムーズにコンテンツを切り替えられるシングルページをベースとし、UI/UX に優れたものである。本論文では、SuperSQL と React を組み合わせることでモダンな Web ページを生成するシステムを提案する。SuperSQL とは SQL の拡張言語であり、独自のクエリを記述し関係データベースの出力結果を構造化することによって多彩なレイアウト表現を実現することのできるものである。本論文の提案するシステムにより、従来よりも少ないコード量で手軽に高度な UI/UX を備えたモダンな Web ページを生成することができ、Web ページ開発者の負担を軽減しながらリッチなユーザ体験を提供することが可能となる。

キーワード Web, UI/UX, React, SQL, SuperSQL

1 はじめに

近年の Web 技術の発展により、Web ページの表現力は高度化している。それに伴い、ページ内の情報だけでなく、ページの見やすさや利用しやすさといった部分も求められるようになっていく。こうした背景から、モダン Web という概念が広がっている。モダン Web では、1 つの Web ページ内に複数のコンテンツを含み、ページの遷移を行うことなくスムーズにコンテンツを切り替えることのできるシングルページが主流となっている。また、高度な UI/UX を備えており、動的なページであるという特徴もある。このようなモダン Web ページはデザインにも優れ非常に魅力的である一方、動きを出す部分が重要になるため、昔ながらの Web ページに比べ作成に知識や手間が要求されるという難点がある。

これを解消する方法として、本論文では SuperSQL [1] [2] と React [7] を組み合わせることでモダンな Web ページを生成するシステムを提案する。SuperSQL とは、独自のクエリを記述することで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実現することのできる SQL の拡張言語である。Web ページを形作る HTML を生成するような場合において、SuperSQL はデータベースとの連携が容易であるだけでなくデザイン面の操作も行うことができるという特徴もあり、非常に利便性が高い。これまでに SuperSQL を用いた Web ページ生成に関する研究は数多く行われてきた [3] [4] が、従来ではレスポンシブなページやデザインテンプレートを使用したページは作成できても、アプリバーやタブを用いた動的なページを作成することはできていなかった。本論文の提案するシステムを実装することにより、少ないコード量で手軽に高度な UI/UX を備えたモダンな Web ページを生成することが可能になる。これにより、Web ページ開発者の負担を軽減しながらリッチなユーザ体験を提供することができる。

本論文の構成は以下の通りである。まず、第 2 章で関連研究及び関連技術について述べる。次に、第 3 章で SuperSQL の概要について述べる。そして、第 4 章で提案手法であるモダン Web 生成システムについて述べる。第 5 章で評価について述べ、最後に第 6 章でまとめを記述する。

2 関連研究・技術

2.1 関連研究

本論文ではモダンな Web ページを従来よりも手軽に生成する方法として SuperSQL を使用することを提案しているが、別のアプローチからコード量の削減を図った研究もある。Naiki ら [5] は、状態遷移図を記述することで動的な Web ページの生成を行った。Web ページの開発者はコンポーネントの状態を表す円と状態間の遷移を表す矢印を描写するだけでよく、トランスレータがこれを自動で React のコードに変換することで Web ページが生成される。この手法ではコード記述が不要であるため、プログラミングの学習において有用である。

2.2 関連技術

2.2.1 React

React [7] は Facebook 社が開発している UI 構築のための JavaScript ライブラリである。React にはコンポーネントベースという特徴があり、カプセル化されたコンポーネントを組み合わせることで高度な UI を構築する。この特徴によって UI を部品に分解して考えることができるようになり、効率的な更新・描画や部品の再利用ができたりするようになる。

React コンポーネントは `render()` メソッドを実装することで作成できる。このメソッドは、受け取ったデータを元にして実際に表示する内容を返すものである。例えば、次のような場合、`HelloMessage` クラスが 1 つのコンポーネントとして扱われている。

React の用例

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello World!
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage />,
  document.getElementById('hello-example')
);
```

シングルページ開発において便利である React は現在大手企業の Web サイト [8] [9] でも用いられており、その有用性は広く知られている。

2.2.2 Material-UI

Material-UI [10] は React 向けの UI コンポーネントのライブラリである。コンポーネントの種類が非常に豊富であり、インポートするだけで簡単に利用することができるという特徴がある。また、用意されているコンポーネントの簡易的なカスタマイズも行えるため、好みのデザインの Web ページを作るのに適している。

3 SuperSQL

本章では SuperSQL について簡単に述べる。SuperSQL のアーキテクチャを図 1 に示す。

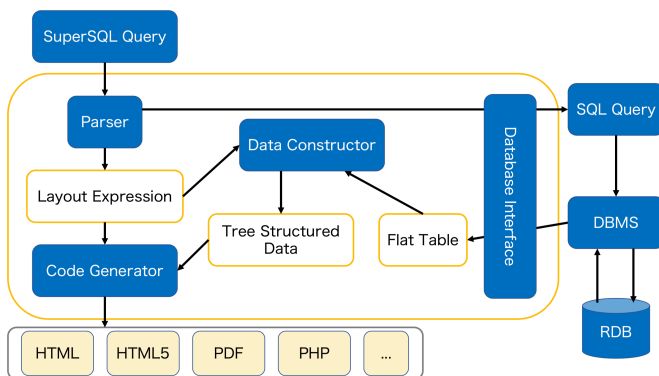


図 1 SuperSQL のアーキテクチャ

SuperSQL は、独自のクエリ記述により関係データベースの出力結果を構造化して多彩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] この独自のクエリは、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>は出力媒体を示す

部分であり、HTML, Mobile HTML5, 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 を表しており、結合子や反復子などといった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

結合子は取得し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 (,)

データを横方向に結合して出力

例 : Name, Age

Name	Age
------	-----

- 垂直結合子 (!)

データを縦方向に結合して出力

例 : Name! Age

Name
Age

- 深度結合子 (%)

データを 3 次元方向に結合して出力

例 : Name % Age

Name	→	Age
------	---	-----

3.2 反復子

反復子は取得したデータをどの方向に繰り返し表示するかを指定する演算子である。括弧内はクエリ中の演算子を示している。

- 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方向に繰り返し出力

例 : [Name],

Name1	Name2	...	NameN
-------	-------	-----	-------

- 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方向に繰り返し出力

例 : [Name]!

Name1
Name2
...
NameN

3.3 装飾子

SuperSQL では、関係データベースから抽出された情報に文字のサイズや位置、横幅、縦幅などといった情報を付加することができる。これらの情報は属性名と装飾演算子 (@) を用いて指定し、

<属性名>@{<装飾指定>}

という形式で記述する。装飾指定は”装飾子の名称=その内容”で指定する。また、”, ”を用いると例えば文字のサイズと色といったように複数の装飾を同時に行うことができる。

3.4 関数

SuperSQL にはいくつかの関数が用意されている。画像を表示することのできる image 関数やリンクを生成することのできる link 関数、クロス表を生成することのできる cross_tab 関数など様々なものがある。

また、通常の間数とは異なる集約関数というものもある。これは、複数の入力行から 1 つの結果となる値を生成する関数である。例としては最大値を返す max, 最小値を返す min, 平均値を返す avg などが挙げられる。集約関数では引数を丸括弧ではなく角括弧で囲う形式で記述する。

4 モダン Web 生成システム

本章では、提案手法であるモダン Web 生成システムについて述べる。

4.1 システム概要

本システムでは、予め定義された Web ページのフレームを呼び出すことで手軽にモダンな Web ページを生成することができる。図 2 にシステムの流れを示す。

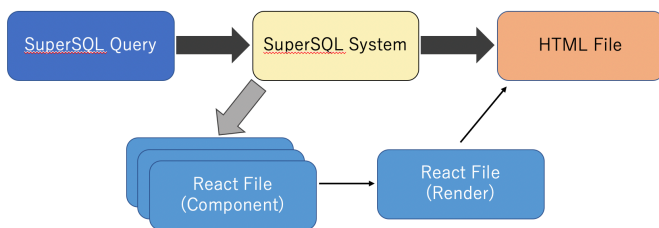


図 2 モダン Web 生成システムの流れ

システムのユーザは、SuperSQL クエリの装飾子を用いて Web ページのフレームを指定し、SuperSQL システムを通して HTML ファイルを生成する。図 2 において、この流れは大きな黒い矢印で示されている。React [7] を用いて書かれたファイルのうちどれを使用するかはフレーム毎に決まっている。ユーザがフレーム内の特定のコンポーネントに対して任意の文字列を入れるなどの簡易的な手入りを施したい場合、そのコンポーネントに対応した SuperSQL の関数機能を用いることで実現することができる。これは大きな灰色の矢印で示されている。小さな黒い矢印はファイルの依存関係を示している。

また、コンポーネントとしてページ内コンテンツ切り替えが

できるタブ機能をフレーム内で使用する場合、フレーム部分とは別にコンテンツ部分を作る必要がある。フレーム部分とコンテンツ部分を連携させることで、ページ遷移を伴わないシームレスなコンテンツ切り替えを行うことのできる Web ページを生成することが可能となる。1 つのフレームに対して複数のコンテンツを持つことができるため、フレーム部分とコンテンツ部分は一対多となっている。

2 つの部分に分けて生成することで、システムユーザが記述する SuperSQL クエリが複雑になりすぎてしまうことを防止できる。それぞれの部分については、4.2 節と 4.3 節でそれぞれ詳しく説明する。

4.2 フレーム部分の生成

フレーム部分を生成する SuperSQL クエリの例を次に示す。

```
GENERATE HTML
{
  appBar('Top Page')!
  tab(t_cont('Main Menu', 'c1.ssql'),
      t_cont('Sub Menu', 'c2.ssql'),
      t_cont('Set Menu', 'c3.ssql'))
}@{frame='template1'}
```

フレーム部分を生成するにあたっては、SuperSQL の装飾子指定を用いることで予め決まったデザインのものを使用することができる。この例のクエリでは、template1 というデザインのフレームを使用している。また、appBar 関数を用いてフレーム内のコンポーネントであるアプリバーに対して 'Top Page' という文字列を入れている。図 3 に生成されたアプリバーを示す。そして、tab 関数を用いてフレーム内のコンポーネントで



図 3 生成された文字列入りのアプリバーの例

あるタブがそれぞれコンテンツ部分と連携できるようにしている。tab 関数の引数には t_cont 関数を取っており、コンテンツ部分を生成する SuperSQL クエリのファイル名と結びつけている。アプリバーやタブといったコンポーネントは UI コンポーネントのライブラリである Material-UI [10] のものをベースとして利用している。

4.3 コンテンツ部分の生成

タブの利用に伴うコンテンツ部分を生成する SuperSQL クエリの例を次に示す。

GENERATE HTML

```
[
  null((asc)m.id)!
  image(m.img, './pics')@{width=200, align='center'}!
  m.name@{width=150, font-size=20, align='center'}],
  m.price||'円'@{width=100, font-size=20, align='center'}
],3!@{table-align='center'}
FROM menu m
```

コンテンツ部分の生成に関しては、従来の SuperSQL クエリと同様の記述を行うだけで良い。ただし、ファイル名がフレーム部分の生成時に tab 関数の引数で使った t_cont 関数のものと一致している必要がある。この例のクエリはデータベースにあるメニューの画像と名前及び値段を表示するものだが、フレーム部分で用いた 1 つ目の t_cont 関数と連携させるためファイル名は c1.ssql としている。図 4 にフレーム部分とコンテンツ部分の連携により生成されたタブを示す。図 4 では Main Menu



図 4 生成されたタブの例

の内容を表示しているが、c2.ssql 及び c3.ssql といったコンテンツ部分のクエリを同様に記述・実行することで、Sub Menu や Set Menu に関してもページを遷移することなくスムーズに表示することが可能である。

5 評価

本章では評価として、提案手法を使用してモダンな Web ページを生成した場合と提案手法を使用せずに生成した場合のコード量を比較し、提案手法の有用性について考察することを予定している。

6 まとめ

6.1 結論

本論文では、近年広まっているモダンな Web ページを生成するにあたり、SuperSQL と React を組み合わせて用いるシステムを提案した。このシステムを利用することにより、Web ページの開発者は従来よりも手軽にモダンな Web ページを生

成することができるように、React などの知識がなくともページ閲覧者に対してリッチなユーザ体験を提供することが可能となる。

6.2 課題・展望

本論文の提案する手法ではモダンな Web ページをフレーム部分とコンテンツ部分に分けて生成しているが、フレーム部分についてはまだ数少ないコンポーネントしか利用することができないのが現状である。今後は更にコンポーネントの種類を増やすことが大きな課題である。そのためには、appbar 関数や tab 関数のように SuperSQL の関数機能を更に充実させることが必要となる。

また、現在の提案システムでは、フレーム部分は既に決まったデザインのものを利用しているに過ぎない。例えばアプリバーに文字を入れるというような簡単な手入れはできるが、コンポーネント自体の色を変えるなどの編集は行うことができない。この部分についても改善の余地があると言える。具体的には、システムユーザが GUI でコンポーネントの追加・編集・削除を行い、テンプレートとなるオリジナルのフレームをデザインできるツールを作成することを考えている。このようなツールを実現することができれば、Web ページ開発者は複雑なコードを記述することなく簡単にフレームデザインを行うことができ、生成できる Web ページの見た目に幅を持たせることが可能となる。

文 献

- [1] SuperSQL: <http://ssql.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] K. Haruno, K. Goto, M. Toyama: "Modern Web Page Generation by Using Exclusively SuperSQL". Proceedings of IDEAS'16: the 20th International Database Engineering Applications Symposium, pp. 340-345, 2016.
- [4] R. Koshijima, K. Goto, M. Toyama: "Generating responsive web pages using SuperSQL". Proceedings of iiWAS'16: the 18th International Conference on Information Integration and Web-based Applications and Services, pp. 231-240, 2016.
- [5] S. Naiki, M. Kohana, S. Okamoto, M. Kamada: "A Graphical Front-End Interface for React.js". Proceedings of the 21st International Conference on Network-based Information Systems, Lecture Notes on Data Engineering and Communications Technologies, vol. 22, pp. 887-896, 2018.
- [6] G. Zhang, J. Zhao: "Visualizing Interactions in AngularJS-based Single Page Web Applications". Proceedings of SEKE 2018: the International Conference on Software Engineering and Knowledge Engineering, pp. 403-408, 2018.
- [7] React: <https://ja.reactjs.org/>
- [8] Airbnb: <https://www.airbnb.jp/>
- [9] Uber: <https://www.uber.com/jp/ja/>
- [10] Material-UI: <https://material-ui.com/>