

SuperSQL における Ajax ページ生成のレイアウト機能拡張

冠木 亮[†] 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: [†]kabu@db.ics.keio.ac.jp, ^{††}toyama@ics.keio.ac.jp

あらまし 従来の SuperSQL での Ajax ページ生成には、一ページに対し一つのフロントエンドしか持つことができない、GUI の種類が少ないなどの制約があった。本論文では、GUI の追加と、複数のフロントエンドを単一のページ上で制御する手法を提案し、Ajax ページ生成におけるレイアウト機能を拡張した。

キーワード SuperSQL, HTML, Ajax, RIA

Layout Function Extends for Generating Ajax Pages of SuperSQL

Ryo KABUKI[†] 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: [†]kabu@db.ics.keio.ac.jp, ^{††}toyama@ics.keio.ac.jp

Abstract For Ajax page generation in conventional SuperSQL, there was the limitation that only one front end could have for one page, and that a kind of the GUI was poor. This article proposes adding GUI and technique to control plural front ends on a single horn page and expanded the layout function in the Ajax page generation of SuperSQL.

Key words SuperSQL, HTML, Ajax, RIA

1. はじめに

近年インターネットの普及と共にウェブに関する様々な技術が向上し、従来の HTML だけでは表現しきれなかった、よりユーザエクスペリエンスなウェブページが増えている。その中でも特に RIA(Rich Internet Application)[5] と呼ばれるウェブアプリケーション、ページは従来のユーザインタフェースの貧弱さや通信量の多さなどといった多くの問題点を解消する新しい技術として注目されている。RIA 作成技術は様々なものが存在するが、特に近年注目を集めているのが Ajax [6] である。

一方、遠山研究室で研究されている SuperSQL はクエリを発行することで HTML や PDF, Excel などにデータベースの内容を出版する技術である。特に出力媒体に HTML を選択した場合に、Ajax 通信を用いたウェブページを生成することができる。しかし、従来の SuperSQL では Ajax 対応ページを生成する際に、一つのページに対して一つのフロントエンドしか持つことができず、また生成可能な GUI が少ないといった制約があった。

本稿では、上記の制約を解消するために単一のページ内で複数のフロントエンドを同時に制御する手法を提案し、また新たな GUI の追加を行うことで SuperSQL の Ajax 対応ページ生成機能の拡張を行った。

以下、本稿の構成を示す。まず 2 章で SuperSQL の概要につ

いて述べる。次に 3 章で従来の Ajax 対応ページ生成機能の制約について述べ、4 章で本論文で提案する手法について述べる。5 章では提案手法の実験と評価について述べる。最後に、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子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)

2.5 Ajax 対応ページの生成

SuperSQL で Ajax 対応ページを生成するには、出力メディアを HTML に指定し、作成オプションに -ajax を付ける。これによって Ajax ページ生成のために用意されたいくつかの装飾子を使用することができるようになり、オンラインで Ajax 通信を利用して SuperSQL サブレットを実行することができるようになる。従来の SuperSQL による Ajax ページ生成機能のシステム概要を図 1 に示す。

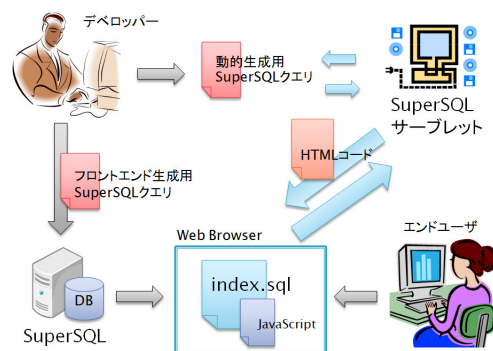


図 1 従来の Ajax 対応ページ生成機能の概要

デベロッパーはフロントエンドとなる index ページを生成するクエリと、コンテンツを生成するクエリ群を作成する。ページの閲覧者であるエンドユーザは index ページにアクセスし、ウェブページを閲覧する。クリックやドラッグアンドドロップなどのユーザインタラクションに応じて index ページは JavaScript を利用して、サーバーを経由して SuperSQL サブレットにリクエストを発行する。サブレットはリクエストのファイル名や条件などを利用して、SuperSQL を起動し、HTML コードを生成する。JavaScript はサブレットからのレスポンスとして HTML コードを受け取り、対象となるウェブページの一部を動的に更

新する。

この従来の SuperSQL による Ajax 対応ページ生成機能の詳細については SuperSQL を用いた RIA 作成支援の提案[4]を参照のこと。

3. 従来手法の制約

3.1 レイアウトの制約

従来の SuperSQL による Ajax 対応ページ生成では、index ページを生成する SuperSQL クエリである index.sql と、動的生成用の SuperSQL クエリ群によって一つのウェブページを構成する。

動的生成用の SuperSQL クエリ群は、ユーザインタラクションに応じてページの一部を更新するためのクエリであり、ページ全体のレイアウトは index ページを生成する SuperSQL クエリによって決まる。従来のシステムは、index.sql ファイルは一つであるという前提のもとで開発されているため、複雑な構造を持ったページを生成するためにはそのレイアウト情報をすべて一つの index.sql ファイルに書き込む必要があった。この index.sql ファイル内では、SuperSQL の関数である embed 関数によって、一つの HTML を複数の div 要素に分割したものを並べ替えてレイアウトを行うため、要素数が増えるにつれクエリが煩雑になり、所望のレイアウトを組むことが困難になっていく。また、一つのウェブページに対し一つのフロントエンドしか持つことができないため、別途作成したフロントエンドをページの一部として利用することができないといった制約があった。

3.2 GUIの種類

GUI については、div 要素を利用したドラッグアンドドロップ、リンク先をポップアップウィンドウで表示するウィンドウ、選択した画像群を一枚ずつ表示させるアルバム、検索フォームなど、Ajax を駆使した基本的な GUI が用意されている。これらは SuperSQL クエリ中に、装飾子として数文字の記述を加えるだけで簡単に実装することができるようになっているが、RIA を生成するにあたって十分であるとはいえない。

3.3 制約の解消

本稿では以上に述べたような制約を解消するため、複数のフロントエンドクエリを同時に制御する手法を提案する。また、GUI の不足を補うために、メニューバーやカレンダーなどの新しい GUI の追加を行う。次節にてその方法について詳しく述べる。

4. 提案手法

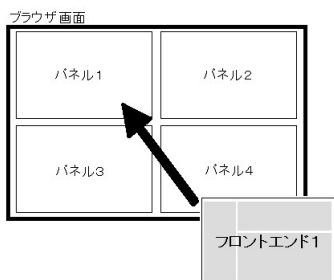


図2 パネルのレイアウト



図3 複数フロントエンドの制御

従来の Ajax ページ生成では、index.sql というクエリにフロントエンドを記述していた。このクエリによって生成されたウェブページが index.html という名前のフロントエンドとなり、ブラウザ上で表示されることになる。ところが全く別のフロントエンドを別名のクエリで用意し、それらを同時に実行したい場合など、一つのページ上に複数のフロントエンドを制御しようとする、index.sql を一つしか持つことができないためうまくゆかない。

そこで、各々がフロントエンドを独立して所持できる局所的なページとしてパネルという概念を導入する。図2は index.sql にてパネルのレイアウトを行う様子を表わす概念図である。index.sql にはこのようにパネルのレイアウトについて記述することとし、レイアウトには SuperSQL の水平、垂直結合子を用いることができる。また、各パネルがどのフロントエンドを持つかを index.sql クエリ内に記述してやることで、配置されたパネルにそれぞれクエリとして用意したフロントエンドを持たせることができ、これによって一つの画面内に複数のフロントエンドを、自由にレイアウトして同時に実行させることができる。図3は index.sql によって4つのパネルを用意し、その一つ一つに別のフロントエンドを配置した時の概念図である。



図4 パネルに対する各領域

また、図4のように、各フロントエンドに対する相対的な領域を考え、これらの各領域に自由に GUI を付加させることができるようにした。たとえば、パネルの上部(図4では north と書かれた部分)にメニューバーを配置し、右部(図4では east と書かれた部分)にはタブを配置する、といったことをフロントエンド単位でできるため、パネルのレイアウトと合わせることでより複雑なレイアウトを持つ Ajax ページを生成できるようになった。

これらの実装を行うにあたり、従来の SuperSQL で使用されていた Ajax フレームワークである Yahoo!UI [9] と上位互換を持つフレームワークである Ext JS [7] を使用した。以下では、SuperSQL 上でこの手法を実現させるために新しく追加した関

数について述べる。

4.1 centerlayout 関数

index.sql において、パネルのレイアウトを行うための関数であり、この関数によってパネルが生成される。ユーザは生成したいパネルの数だけこの関数をクエリに書き込む必要があり、centerlayout 関数同士を SuperSQL の水平、垂直結合子で結ぶことによってパネルのレイアウトを行う。

centerlayout 関数が取ることのできる引数は、表 1 の通りである。ここで、引数 file に sql ファイルが指定された場合は、SuperSQL サーブレットを起動して結果を HTML コードとして取得してパネルに反映させる。引数 file に html ファイルが指定された場合は、そのファイルの内容をパネルにそのまま読み込む。また、引数 file で値を指定しなかった場合、そのパネルには index.sql に書かれたクエリがそのまま実行される。

引数	説明
id	パネルを一意に決定するための識別子
width	パネルの幅
height	パネルの高さ
file	そのパネルに配置したいフロントエンドが書かれた sql ファイル名
cond	file で指定した sql ファイルを実行する際に Where 句に追加する条件式

表 1 centerlayout 関数の引数

4.2 extlayout 関数

centerlayout 関数によって生成されたパネルに対し、GUI を付加する関数である。この関数一つに対し GUI 一つが対応しているため、付加したい GUI の数だけこの関数を記述する必要がある。また、現在の仕様としてすべての extlayout 関数は、centerlayout 関数よりも前に記述する必要がある。extlayout 関数が取ることのできる引数は、表 2 の通りである。

引数	説明
id	GUI を一意に決定するための識別子
cid	GUI を付加させたいパネルの id. 省略した場合は画面全体に対して GUI を付加させる
region	cid で指定したパネルのどの領域に GUI を付加させるか
type	GUI の種類. menu, tab, calender を指定できる
width	GUI の幅
height	GUI の高さ
title	GUI に付ける見出しの文字列
collapse	付加する GUI を折りたためるかどうか

表 2 extlayout 関数の共通引数

また、引数 type の値をメニューバーを表示させるという意味の menu にした場合は、表 3 の引数を、type の値をカレンダーを表示させるという意味の calender にした場合は表 4 の引数を、さらに指定する必要がある。ここでも引数 file には sql ファイルか html ファイルを指定することができ、それぞれの挙動

は centerlayout 関数で説明したものと同じである。また、type に menu を指定したときの表 3 の各引数は、メニューバーに表示させたいボタンの数だけ指定する必要がある、それぞれの引数の後ろにボタンの数に合わせて N(N=1,2,3...) を付けて指定する。例えば、メニューバーに 3 個のボタンを表示させたい場合は引数 obj を、obj1, obj2, obj3 のように 3 個記述することになる。

引数	説明
objN	メニューバーに表示させるボタンのテキスト
fileN	ボタンがクリックされたときに実行される sql ファイル
condN	file で指定した sql ファイルを実行する際に Where 句に追加する条件式
targetN	実行結果を反映させるパネル id

表 3 type に menu を指定したときの引数

引数	説明
file	日付がクリックされたときに実行される sql ファイル
cond	file で指定した sql ファイルを実行する際に Where 句に追加する条件式
target	実行結果を反映させるパネル id

表 4 type に calender を指定したときの引数

4.3 諸問題の解決

4.3.1 スタイルの競合

SuperSQL では、装飾子による部分的なスタイルの指定が可能である。装飾子によって指定されたスタイルは、SuperSQL によって自動的に振られた id によって管理される。この id は一つの sql ファイル内では重複しないように振られるが、本稿で提案した複数フロントエンドを同時に実行する環境では、このスタイル id がクエリ間で競合を起こし、後から読み込まれたスタイルで前に読み込まれていたスタイルを上書きしてしまうという問題が起こった。ここで、一つのファイル内ではスタイル id が重複しないことに着目し、スタイル id の後ろにファイル名を付加することによって、複数のフロントエンドを同時に実行してもスタイルの競合が起こらないように SuperSQL の内部仕様を変更した。

4.3.2 ウィンドウ GUI

従来の SuperSQL には Ajax ページ生成時に使用できるウィンドウ GUI があった。これは、link 関数の装飾子として panel=yes という指定を行うことでリンク先が新しいウィンドウに表示される、といったもので、スタイルシートによって自由に見た目を変更できるという利点があった。しかし、このウィンドウ GUI は表示させたい内容によって著しくデザインが崩れてしまったり、リンク先がうまく表示されない場合があるなど、動作が不安定であった。そこで、デザインなどは変更できないものの、安定してリンク先を表示できるウィンドウ GUI を新たに実装した。ユーザは link 関数の装飾子として extwin=yes を指定するだけでこのウィンドウ GUI を使用することができる。二つのウィンドウ GUI は働きは同じであるものの、どちらにもメリット、デ

メリットがあると考え、どちらを使用するかはユーザが都度選択できるようにした。

5. 実験と評価

5.1 実験方法

提案した手法による Ajax ページ生成の有用性を評価するため、評価実験を行った。被験者は従来の SuperSQL による Ajax ページ生成の経験がある者、従来の SuperSQL による Ajax ページ生成の経験はないものの、普段から SuperSQL を使い慣れている者、SuperSQL についての知識はあるものの、ほとんど使用した経験のない者の 3 つのグループに分け、各グループは 3 名ずつとする。

被験者は簡単な説明を受けた後、20 分程度の時間で自由に提案インターフェースを使用する。後に被験者はアンケート形式で 3 項目について、5 点 (そう思う)、4 点 (どちらかというと思う)、3 点 (どちらとも言えない)、2 点 (どちらかというと思わない)、1 点 (そう思わない) の 5 段階で評価する。項目は以下の通りである。

- Q1. 思い通りの出力が得られたか
- Q2. クエリの書き方はわかりやすいか
- Q3. Ajax 通信は想定通りに動作したか

また、Ajax 対応ページを生成するために必要なコード量の比較により、本稿での拡張の有用性を検討した。メニューバーによるページの遷移を行うだけの単純なレイアウトの Ajax 対応ページと、ウィンドウ GUI やカレンダー、汎用パネルを駆使した複雑なレイアウトの Ajax 対応ページを、それぞれ本稿で拡張した SuperSQL を使用して生成した場合と、すべて手書きで同様のページを生成した場合の、必要となる各コード量を比較した。単純なレイアウトの Ajax 対応ページが生成された様子を図 5 に、複雑なレイアウトの Ajax 対応ページが生成された様子を図 6 に示す。

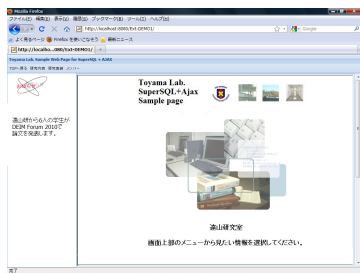


図 5 簡単なレイアウトのページ

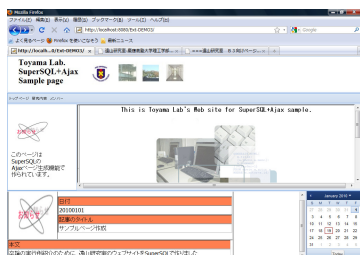


図 6 複雑なレイアウトのページ

5.2 実験結果と評価

アンケートを取る実験の結果を図 7 に示す。この図において、グループ A とは従来の SuperSQL による Ajax 対応ページ生成の経験のある者を、グループ B とは普段から SuperSQL を使用しているものの、Ajax 対応ページ生成の経験はない者を、グループ C とは、SuperSQL の基本的知識はあるが、ほとんど利用経験のない者をそれぞれ表す。

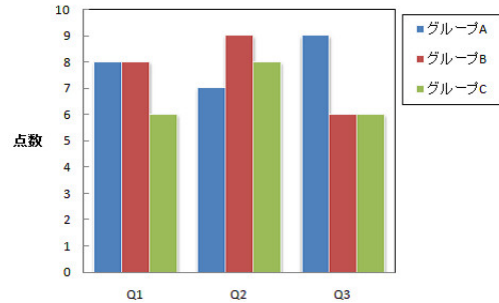


図 7 アンケート結果

Q1. の思い通りの出力結果が出たかについては、Ajax ページ生成の機能を使ったことがあるかどうかでは差が出ない結果となった。一方で、SuperSQL そのものに使い慣れていないグループ C では点数が低くなった。これは、本システムにおいてパネルのレイアウトは SuperSQL の結合子を使って行うことができるため、SuperSQL で複雑な表が自由に生成できるユーザには Ajax ページを生成する際のパネルのレイアウトも問題なく行えるからであると考えられる。

Q2. のクエリの書き方については従来の Ajax ページ生成機能を利用したことがあるグループ A で低い点数となった。従来のシステムでは、ページ内のレイアウトはすべてフロントエンドを生成するためのクエリ index.sql に書き込む必要があり、その際は別途用意した html ファイルや SuperSQL クエリなどを SuperSQL の embed 関数によって結合して行うのに対し、本システムでは GUI の追加は各パネルに対する相対的な領域を指定することで行うといった方法を取っている。この方法は直感的に GUI の配置ができるため、SuperSQL に慣れていなくても既存の SuperSQL クエリに数行記述するだけで簡単に GUI のレイアウトをすることができると言える。反面、従来システムに慣れているユーザにはわかりやすい書き方であるとは言えないという結果となった。

Q3. の Ajax 通信に関する設問に対しては、やはり従来システムで Ajax ページの生成を経験したことがあるグループ A では点数が高く、SuperSQL には慣れていても Ajax 通信を使用したことがないグループ B と、SuperSQL そのものに慣れていないグループ C では点数は低くなっている。本システムでの記述方法では、極力 Ajax 技術を意識せずに使えるようになっているが、パネル ID によるターゲット指定や、動的に実行される SuperSQL クエリの用意などが必要になるため、SuperSQL に慣れているユーザでも Ajax ページ生成機能の習熟度による影響は多少受けてしまうことがわかる。

また、コード量の比較を行う実験の結果として、簡単なレイア

ウトのページを生成した場合のコード量を表 5 に, 複雑なレイアウトのページを生成した場合のコード量を表 6 に示す.

知識	手書き	本システム
JavaScript	38 行	-
Ext JS	47 行	-
HTML	176 行	17 行
SuperSQL	-	67 行
合計	261 行	84 行

表 5 簡単なレイアウトのページ生成に必要なコード

知識	手書き	本システム
JavaScript	52 行	-
Ext JS	87 行	-
HTML	674 行	50 行
SuperSQL	-	80 行
合計	813 行	128 行

表 6 複雑なレイアウトのページ生成に必要なコード

SuperSQL を使用することで手書きで HTML のソースコードを書くコストが大幅に削減できるだけでなく, 本システムの利用によって, JavaScript や Ext JS の知識を全く必要とせず自由にレイアウトを組んだ Ajax 対応ページを生成できることが大きな利点であると言える. 表 5 と表 6 を比較すると, ページのレイアウトがある程度複雑になっても必要な SuperSQL クエリの行数には大きな変化はないことがわかる. このことから, フロントエンドのレイアウトを決定する index.sql は簡単なレイアウトでも複雑なレイアウトでもわずかな記述で実現できるということが言える. すべてを手書きで表現しようとする場合, GUI が増えればその分だけ必要となる Ext JS ソースコードの量も増えていくため, 複雑なレイアウトを実現しようとするればするほど本システムが有効に働くと考えられる. またこの結果から, HTML のソースコード量が他の項目に対して大きく異なっていることがわかるが, これはウェブサイトを使用するデータベースの情報量が増えるほど SuperSQL が生成する HTML ソースコードが増えていくことによると考えられ, HTML ソースコードのコスト削減という面では本システムよりも SuperSQL そのものによるところが大きいと考えられる.

6. ま と め

本研究では従来の SuperSQL での Ajax 対応ページ生成において, 一つのページに対し一つのフロントエンドしか持つことができない点, GUI の種類が少ない点などの制約を解消した. 同時に, これらの改善に伴って発生した SuperSQL の諸問題を解決し, より自由なレイアウトの Ajax 対応ページを SuperSQL から生成できるようにした.

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] 遠山 元道: 『ターゲットリストの拡張によるデータベース出版

と概視の実現』, 信学技報, Vol.93, No.152, P79-88, 電子情報通信学会, 1993

- [4] 小林 武彦, 遠山 元道: 『SuperSQL を用いた RIA 作成支援の提案』信学技報, Vol.107, No.131, P289-294, 電子情報通信学会, 2007
- [5] Jeremy Allaire.: "Macromedia Flash MX-A next-generation rich client. " Technical report, Macromedia, March 2002
- [6] Linda D. Paulson : "Building Rich Web Applications with Ajax. " *Computer*38(10) , pp.14-17, 2005
- [7] ExtJs: <http://extjs.co.jp/>
- [8] 古籟一浩, 石丸健太郎: 『ExtJS 入門 リッチ UI な Web サイトをつくる Ajax フレームワーク』 秀和システム, 2009
- [9] YAHOO! UI Library: <http://developer.yahoo.com/yui/>