

〔論 文〕

パンフレット・ウェブ・アプリケーション生成ツール Pamphwag について

山田 貴裕^{*1}

Pamphwag: A Pamphlet Web Application Generator

Takahiro YAMADA^{*1}

Abstract

Pamphwag, a pamphlet web application generator, was developed to allow users to easily create pamphlet apps from pamphlet data without the requirement of any programming skills. The store-search function is important for pamphlet apps. The pamphlet apps generated using Pamphwag have a function for exploring stores using various conditions such as keywords and location information. An attempt was made to generate a real pamphlet app with Pamphwag, and the app was successfully generated.

Key Words : Web Application, Pamphlet Application, Application Generator

1. 背景と目的

久留米市には「久留米市地産地消推進店登録制度」があり、その取り組みとして久留米市の地産地消推進店を紹介したパンフレットを作成し、配布している。このパンフレットには、「久留米市地産地消推進店登録制度」に登録されている全ての店舗の情報が掲載されている。久留米工業大学では、久留米市の依頼を受け「久留米市地産地消推進店」パンフレットをスマートフォン向けのアプリにした「久留米市地産地消推進店」アプリ^①の開発が行われていた。「久留米市地産地消推進店」アプリは Android アプリとウェブ・アプリの2つの形態で提供されていたが、Android アプリはウェブ・アプリを Adobe PhoneGap で Android アプリ化したものなので、開発方法はウェブ・アプリと同等だった。

また、久留米市保健所には「くるめ健康づくり応援店制度」があり、その取り組みとして「くるめ健康づくり応援店ガイドブック」を作成し、配布している。このガイドブックにはくるめ健康づくり応援店の登録店や健康を維持するための食事のポイントが掲載されている。久留米市保健所からも「くるめ健康づくり応援店ガイドブック」をアプリ化する依頼があった。このアプリ化については登録店の紹介だけではない部分もあったが、ほぼ同様の仕組みで実現できると考え、「くるめ健康づくり応援店」アプリの開発も行うことにした。

ところで、「久留米市地産地消推進店」パンフレットや「くるめ健康づくり応援店ガイドブック」をアプリ化する最大の理由は、パンフレットのような印刷物では常に持ち運ぶことは面倒だが、スマートフォンのアプリであればいつでも利用でき、利便性が向上するためである。これらのパンフレットやガイドブックと同様の印刷物は多数存在しているため、そのようなパンフレット等のアプリ化の需要もあると考えられる。以後、このようなパンフレット等をアプリ化したものをパンフレット・アプリと呼ぶ。パンフレット・アプリを開発するにはソフトウェア開発に関する専門的な知識が必要となるため、パンフレットを作成している者がパンフレット・アプリまで開発することは困難である。しかし、パンフレットのデータから簡単にパンフレット・アプリを生成できるツールがあればパンフレットのアプリ化は容易になる。そこで、「くるめ健康づくり応援店」アプリの開発に先立って、パンフレット等のデータからパンフレット・アプリを生成するツールを開発することにした。

生成するアプリケーションは「久留米市地産地消推進店」アプリがウェブ・アプリをベースにしていたこともあり、ウェブ・アプリとした。ウェブ・アプリであればプラットフォームに依存せずに利用でき、利用者はインストールする手間もかけずに利用できるという利点がある。パンフレット・アプリの処理は軽量なので、グラフィックスを駆使した

^{*1} 情報ネットワーク工学科
令和元年10月31日受理

ゲーム等とは異なりスマートフォンの性能が問題になることはない。また、Adobe PhoneGap 等を利用すれば簡単にネイティブアプリ化することも出来る。本研究において開発するツールはパンフレット・ウェブ・アプリケーションを生成するツールなので、Pamphlet Web Application Generator から Pamphwag と命名した。Pamphwag は利用における技術的な敷居を低くするため、プログラミングが一切不要なツールとして開発することにした（ここでのプログラミングには HTML や CSS を記述することは含まれない）。

ところで、プログラミング不要でアプリケーションを生成できる製品は幾つか存在している。例えば Web Performer⁽²⁾ はプログラミングを必要とせずにデータベースを駆使した複雑な業務システムを生成できるようである。同様に業務システムを生成するための製品は他にもあるようだが、それらの製品は業務システムの開発工程を短縮するためのものであり、それらの製品を用いてアプリケーションを開発する際にシステム開発の知識が不要となるわけではない。また、ウェブ・カタログを作成するためのツールもウェブ・サービスとして幾つか存在しているが、それらのサービスは幾つかの製品やサービスのカタログを作成するためのものであって、本研究で呼ぶところのパンフレット・アプリを作成できるようなサービスではない。

本研究の目的は、プログラミング等のソフトウェア開発に関する専門的な知識を必要とせずにパンフレット・アプリを生成するツールを開発し、これを提供することでパンフレット等のデータから容易にパンフレット・アプリを作成できるようにすることである。ただし、Pamphwag が生成するアプリはウェブ・アプリケーションなので、ウェブ・デザインの知識は必要となる。

2. 生成できるウェブ・アプリケーション

Pamphwag によって生成されるウェブ・アプリケーションはクライアントのみで動作する。即ち、必要なリソースがローカルファイルとして存在していれば、ウェブブラウザで index.html を開けば使用できるようになっており、サーバと連携するような機能はない。勿論、ウェブサーバ上にリソースとして配置しておけば、通常のウェブページと同様にアクセスできる。パンフレット・アプリとして必要なデータは1つの JavaScript ファイルに保存されていて、script 要素でそのスクリプトを読み込むことでクライアント上に保持するようになっている。ただし、画像や動画等のように通常は独立したファイルとして扱うようなデータはサーバ上にファイルとして配置しておき、必要な時に読み込むようになっている。このような動作は通常のウェブページと同様である。画像のようなデータを除けば、パンフレット・アプリとして必要なデータは文字情報のみであり、パンフレットに掲載される店舗数は数百件程度なので、JavaScript ファイルによってクライアント上に保持されるデータサイズは1 MByte に満たない程度と予想でき、PC やスマートフォンにとって問題になるようなサイズではない。パンフレット・アプリとして必要なデータをまとめて読み込んでしまうので、サーバと連携せずにアプリとしての動作が可能であり、パンフレット・アプリを公開する際にはデータベースサーバ等は必要なく、ウェブサーバがあれば良い。そのため、パンフレット・アプリの公開に際して特に専門的な知識は必要ない。

Pamphwag によって生成されるウェブ・アプリケーションは3～5の画面を有する。必ず有する画面は一覧画面、情報画面、絞り込み画面であり、オプションとして地図画面を有することが出来る。また、スタート画面も生成される

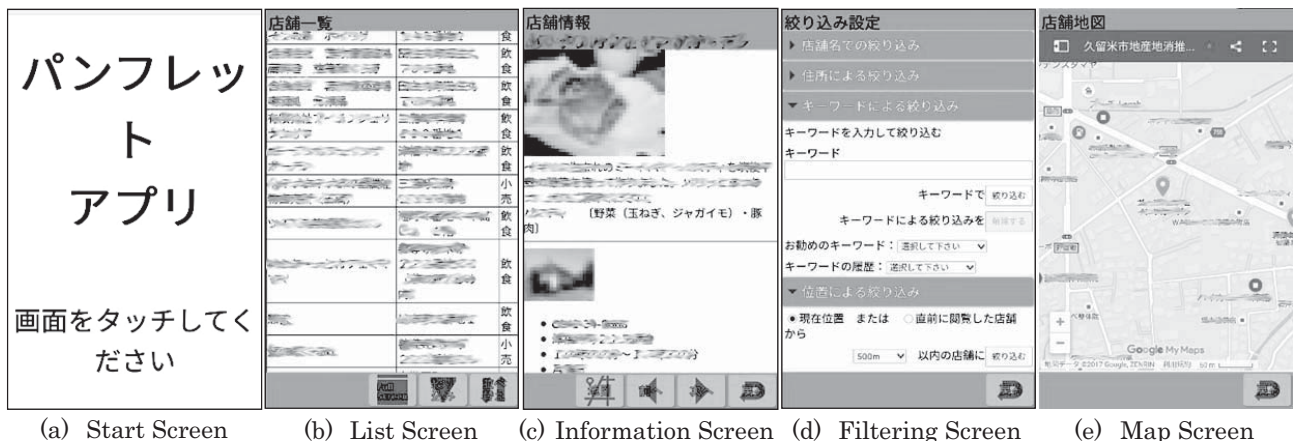


Fig. 1 Screen Examples of Pamphlet App

が、これは削除しても動作に支障はない。本節ではこの画面構成に沿って Pamphwag で生成できるウェブ・アプリケーションの概要について述べることにする。図 1 にパンフレット・アプリの画面例を示す。

スタート画面 (図 1(a)) はパンフレット・アプリを起動した際に最初に表示される画面で、ネイティブアプリのスプラッシュスクリーンや紙媒体のパンフレットの表紙に相当するものである。無くてもアプリとしての動作に支障はなく、削除しても問題ない。スタート画面をタップまたはクリックすると一覧表画面に切り替わる。スタート画面を削除した場合は一覧表画面が最初の画面になる。

一覧表画面 (図 1(b)) は店舗の一覧表を表示した画面である。ユーザはこの画面で閲覧したい店舗を選択する。また、絞り込みを行うには「絞り込み」ボタンから絞り込み画面に移動して設定を行う。更に「並べ替え」ボタンによって一覧表を並べ替えることもできる。「並べ替え」ボタンをタップまたはクリックすると、並び替えのキーとなる項目のメニューが表示される。メニューの項目をタップまたはクリックすると、一覧表が並び替えられる。

情報画面 (図 1(c)) は一覧表画面で選択された店舗の情報を表示する画面である。「前へ」ボタン / 「次へ」ボタンにより、一覧表における前の店舗 / 次の店舗の情報に切り替えることが出来る。地図画面を有する場合は「地図」ボタンから地図画面に移動することが出来る。

絞り込み画面 (図 1(d)) は一覧表画面に表示する店舗を絞り込むための設定を行う画面である。数百の店舗の一覧表から閲覧したい店舗を探すのは面倒な作業になる。そのため、いくつかの条件を設定して一覧表画面に表示する店舗を絞り込むことが出来るようになっている。また、絞り込まれた結果は情報画面の「前へ」ボタン / 「次へ」ボタンの動作にも反映される。絞り込みの機能としては名称による絞り込み、住所による絞り込み、キーワードによる絞り込み、位置による絞り込み、選択肢で選択した値に一致するものに絞り込む機能及び指定した項目が存在するものみに絞り込む機能を実装できる。

地図画面 (図 1(e)) は Google Maps や OpenStreetMap の地図を表示する画面である。この画面を実装するには Google Maps か OpenStreetMap で地図を作成しておく必要がある。また、緯度・経度の情報が情報画面等で表示されることはなくとも、地図画面で店舗を地図の中心に表示するためには店舗の緯度・経度の情報が必要である。

3. Pamphwag の開発

ウェブ・アプリケーションを生成するツールは、生成する際にコンパイルを必要としないため、特定のプラットフォームに依存する必要はない。ツールの利用開始に当たって、そのツールをインストールする必要があると、マルウェアの懸念等により導入のための敷居は高くなる。その点、ウェブ・アプリケーションであればウェブブラウザ上で動作するためインストールの必要もなく、マルウェアについても殆ど心配する必要はない。更に、サーバにデータをアップロードすることなく完全にウェブブラウザ上で動作すれば、ユーザ登録等も必要なく利用できる。そのため、Pamphwag 自体もウェブブラウザ上のみで動作するウェブ・アプリケーションとして開発することにした。開発に当たっては店舗データの形式やユーザインターフェース、店舗の検索機能等について検討した。

3・1 店舗データ

Pamphwag の開発に際し、最初に生成対象とするパンフレット・アプリは「くるめ健康づくり応援店」アプリとすることを目標にした。また、本研究を開始するきっかけになったのは「くるめ健康づくり応援店」だけでなく、その前に開発していた「久留米市地産地消推進店」アプリがあったからであるが、Pamphwag で生成するパンフレット・アプリは「久留米市地産地消推進店」アプリの仕組みを踏襲するものとした。「久留米市地産地消推進店」アプリでは紙媒体のパンフレットは既に存在しており、久留米市地産地消推進店を管理するための表形式の店舗データも存在していた。その為、「久留米市地産地消推進店」アプリでは CSV 形式の店舗データから JavaScript で直接扱える形式のデータに変換して店舗データを利用していた。くるめ健康づくり応援店についても既にハンドブックは存在しており、久留米市保健所によると元となるデータから表形式の店舗データは用意できるとのことだったので、Pamphwag に与える店舗データの形式は 1 店舗当たり 1 行の CSV 形式とした。店舗データにはパンフレット・アプリの情報画面で表示される全ての項目に対応する列が含まれていなければならない。更に地図画面を持つ場合には店舗の緯度・経度も必要であるし、店舗名称で並び替えるには店舗名称の読み方も必要である。

3・2 ユーザインターフェース

生成できるアプリケーションを限定した生成ツールであっても、幾らかのプログラミングを必要とするように設計す

れば、生成可能なアプリケーションの自由度は大きくなる。しかし、本研究の目的は全くプログラミング能力を有しない者でもパンフレット・アプリを作成できるようなツールを提供することであり、その為にはプログラミングのスキルを一切必要としないツールとして Pamphwag を開発する必要がある。その為、生成可能なアプリケーションの自由度は制限される仕様とした。例えば、ユーザインターフェースの形式は HTML 及び CSS を変更する自由度はあるが、ボタンをクリックするとメニューが表示されるといった動作については変更できない。

また、パンフレット・アプリの画面構成は前節で述べたように 3～5 の画面に固定され、画面遷移の方法等も変更することはできない。これは次の理由による。パンフレット・アプリは 3～5 の画面を切り替えて表示するが、これらの画面を別々の HTML ファイルとして作成してはページ間の連携に面倒な仕組みが必要になる。何故なら、ウェブページのプログラムはページに組み込まれているため、別のページに移動すると移動先から移動前のプログラムの変数を参照することはできない。そのため、複数のページに跨って動作するプログラムを作成するには、ページ間でのメッセージのやり取りやデータの保存など、様々な工夫が必要になるからである。そこで、1つの HTML ファイルで複数の画面を表示するようにした。具体的には、1つの画面毎に要素を分け、表示したい画面の要素の style 属性には "display : block;" を指定し、それ以外の画面の要素の style 属性には "display : none;" を指定することで複数画面の 1つだけを表示するようにした。画面の切り替えは、クリックすると別の画面に切り替わるような要素に style 属性を変更するイベントリスナを設定することで行っている。このようにプログラムによって画面遷移を実現しているため、画面の数は 3～5 に固定され、画面の遷移も固定されたものになる。具体的な画面遷移は図 2 に示すようになっている。

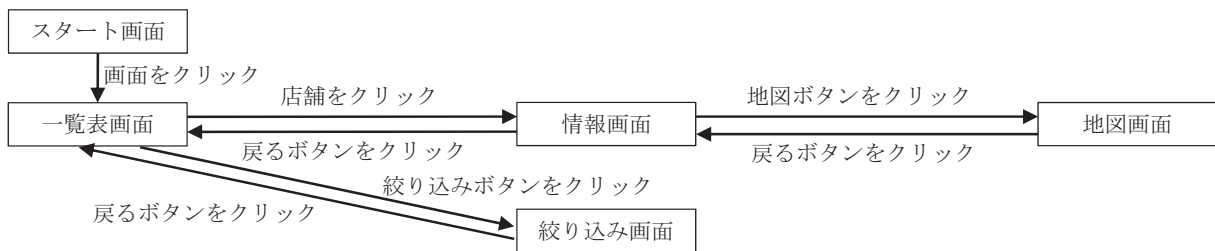


Fig.2 Screen Transitions

ところで、パンフレット・アプリに求められる機能は店舗の情報を閲覧する機能と店舗を検索する機能であり、画面遷移が固定されることは本質的な問題ではないと考える。しかし、パンフレット・アプリだけでなくより応用範囲の広いアプリ生成ツールとするには画面遷移の自由度についても検討する必要があるだろう。

3・3 店舗検索に必要な機能

パンフレット・アプリでは掲載されている店舗をユーザが見つけ易くすることが重要であり、その為の機能としては一覧表における並び替え機能と絞り込み機能が考えられる。一覧表の並び替えは表に表示している項目だけでなく、表形式の店舗データの任意の列をキーとして並び替えることは可能ではある。しかし、キーの候補として店舗データの全ての列の項目を表示すると却って使い難くなってしまふ。そのため、Pamphwag ではどの列の項目を並び替えのキーとするかをアプリ生成時に指定させることにした。一覧表画面の並び替えボタンをクリックすると指定された項目がメニューとして表示され、項目をクリックすると並び替えが実行される。並び替えは昇順と降順がトグルするようにしている。並び替えは指定された列の値の単純な大小比較で行われるため、店舗名称での並び替えには注意が必要である。一覧表画面や情報画面で表示されなくても店舗名称の読み方がデータとして必要であり、キーとしては読み方を指定しておく必要がある。

ユーザが容易に店舗を検索するには条件を指定して一覧表に表示する店舗を絞り込む機能が必要である。Pamphwag では店舗名での絞り込み、住所での絞り込み、キーワードでの絞り込み、位置による絞り込み、その他の絞り込みの機能を実装出来るようにしている。

(1) 店舗名での絞り込み

店舗名を入力することで絞り込むのであれば、キーワードによる絞り込みで可能なので店舗名での絞り込み機能は設ける必要はない。しかし、文字入力を必要とせず店舗名で絞り込む機能を提供しようと考え、店舗名の読みが行、か行…ら行、「やゆよわをん」、ローマ字他で始まる店舗に絞り込む機能を実装できるようにした。この機能を実装する

には店舗データに店舗名称の読み方が必要である。

(2) 住所での絞り込み

住所での絞り込みも店舗名での絞り込みと同様に住所の文字列を入力するのではなく、住所の一覧でチェックを付けた町名のみ絞り込む機能を実装できるようにしている。この機能を実装するには4(7)で述べるように住所（町名等）の一覧表が必要である。なお、1つの市内には100以上の町名が存在することもあるため、あ行～わ行のタブに分けて町名の一覧を表示できるようにしている。タブに分けて表示するには住所の一覧表において、1列目が「あかさ…わ」のどれか1文字で、2列目が町名となっているCSV形式のデータとなっている必要がある。

(3) キーワードによる絞り込み

キーワードによる絞り込みではアプリの生成時に、複数のキーワードが指定された場合のマッチング方法を「どれか含む」「全て含む」「切り替え可能」から選択出来るようにしている。電話番号や緯度・経度等の項目はキーワードによる絞り込みの対象になるとは考え難いので、マッチングの対象は全てのデータとするのではなく、対象となる項目の列番号を指定するようしている。列番号はカンマで区切って複数指定できる。

更に、アプリ制作者がお勧めのキーワードを表示する機能も実装できる。これにより、例えば地産地消推進店アプリでは地域の特産品をお勧めキーワードとして表示するようなことが可能になる。また、キーワードの履歴を表示することも出来るようにした。キーワードの履歴の保存にはブラウザの localStorage を使用する。

(4) 位置による絞り込み

位置による絞り込みは現在位置又は直前に閲覧した店舗の位置からの直線距離によって絞り込む機能である。この機能を実装するには緯度と経度の情報が必要である。位置による絞り込みのためには緯度・経度から2点間の距離を求める必要があるが、パンフレット・アプリでは厳密な計算は必要ないと考える。位置による絞り込みは現在位置又は直前に閲覧した店舗の近所にどのような店舗があるかを確認出来るようにするための機能なので、指定される直線距離はせいぜい10km程度までであり、その範囲で10m程度の誤差があったとしても許容されるだろう。厳密な計算のためにアプリの反応が遅くなる方が問題である。緯度・経度から2点間の距離を求める計算式は幾つか存在するようだが、比較的単純な計算式としてはヒュベニ (Hubeny) の公式が知られている。ヒュベニの公式を以下に示す。

d_y : 2地点の緯度の差 (ラジアン)

d_x : 2地点の経度の差 (ラジアン)

a : 2地点の緯度の平均

R_x : 赤道半径

R_y : 極半径

$E = \sqrt{1 - \left(\frac{R_y}{R_x}\right)^2}$: 離心率

$W = \sqrt{1 - E^2 \cdot \sin^2 a}$

$N = \frac{R_x}{W}$: 卯酉線曲率半径

$M = \frac{R_x(1 - E^2)}{W^3}$: 子午線曲率半径

$D = \sqrt{(d_y \cdot M)^2 + (d_x \cdot N \cdot \cos a)^2}$: 2地点間の距離

ヒュベニの公式は赤道半径と極半径の違いから地球を楕円体として計算するものになっているが、10km程度の範囲ではその違いの影響は小さいと思われる。そこでこの式の赤道半径と極半径にその平均値を与えると、以下の式になる。

$R = \frac{R_x + R_y}{2}$: 赤道半径と極半径の平均

$D = R \cdot \sqrt{d_y^2 + (d_x \cdot \cos a)^2}$: 2地点間の距離

WGS84より赤道半径 R_x を6,378,137.000m、極半径 R_y を6,356,752.314mとして、ヒュベニの公式による計算結果と単純化した式による計算結果とでは、それらの差は久留米工業大学～ヤフオクドーム（福岡市）間40.08kmで48mであり、単純化した計算でも問題はないと判断し、採用することにした。

(5) その他の絞り込み

Pamphwag では、その他の絞り込み機能として選択肢型の絞り込みと○×型の絞り込みを実装出来るようにしている。選択肢型の絞り込みは、複数の選択肢とそれに対応する値を指定して、マッチング対象となる列の値と選択肢の値が一致するものみに絞り込む機能である。選択肢となる部品はチェックボックス、ラジオボタン、選択ボックスから選んで実装するようになっている。複数の選択肢に対してマッチング対象となる列は1つのみである。これにより、例えば図3に示す店舗のカテゴリ（小売店や飲食店、加工食品販売等）で指定したカテゴリの店舗のみに絞り込むような機能を実装することが出来る。

例) カテゴリ

<input type="checkbox"/> 小売店	<input checked="" type="checkbox"/> 飲食店	<input checked="" type="checkbox"/> 加工食品販売	<input type="checkbox"/> ホテル	<input type="checkbox"/> その他	<input type="button" value="設定"/>	<input type="button" value="解除"/>
------------------------------	---	--	------------------------------	------------------------------	-----------------------------------	-----------------------------------

Fig. 3 Choices Filtering

○×型の絞り込みは、主に指定した項目の有無によって絞り込む機能を提供するためのものである。使用する部品はチェックボックスであり、チェックボックスのラベルとその値、マッチング対象の列の組み合わせを指定して実装するようになっている。チェックされている要素の値と列の値が一致するものみに絞り込まれる。選択肢型の絞り込みとの違いは、1つのチェックボックスに対してマッチング対象となる列も1つであることである。ただし、例えば図4に示す栄養表示のように複数の項目を指定して一纏めの絞り込みとすることが出来る。

例) 栄養表示

エネルギー	たんぱく質	脂質	炭水化物	塩分
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 4 OX Filtering

4. Pamphwag によるアプリの生成

Pamphwag は3で述べたようにウェブ・アプリケーションとして開発され、既に以下の URL で公開されている。ここでは公開中の Pamphwag を用いたパンフレット・アプリの生成手順について説明する。

<http://pamphwag.kurume-it.ac.jp>

Pamphwag でパンフレット・アプリを生成するには、表形式の店舗データを用意する必要がある。このデータは3・1で述べたように CSV 形式のファイルでなければならない。また、店舗の画像を使用する場合は別途それらのデータも必要である。そして以下に述べる手順の(1)から順番に設定していく必要がある。以下の説明ではリスト1に示すデータを用いた具体例を挙げる事が出来る。

List 1 Example of shop data

"No", "店舗名称", "読み", "住所", "画像", "カテゴリ", "特典有無"
 1, "○×商店", "まるはつしょうてん", "○×町一丁目16番地○×ビル4階", "img001.jpg", "小売店", ""
 2, "□○食堂", "しかくまるしよくとう", "□○町三丁目1番地", "img002.jpg", "飲食店", "○"
 3, "△□ホテル", "さんかくしかくほてる", "××町1234番地", "img003.jpg", "ホテル", ""
 ...

(1) アプリケーション名称の設定

パンフレット・アプリの名称をテキストフィールドに入力し、「設定」ボタンをクリックする。

(2) 店舗データの読み込み

リスト1に示すような店舗データの CSV ファイルを指定する。ところで、CSV のフィールド区切り記号はカンマ(,) であるが、Pamphwag ではカンマ、タブ、セミコロン (;)、コロンの (:)、半角スペースから選択可能になっている。基本データの欄に CSV ファイルを指定して、フィールドの区切り記号を選択したら「基本データの設定」ボタンをク

リックする。すると“このリンク（アプリで使用するデータ）をクリックして、ファイルを保存して下さい。...”と表示される。このリンクをクリックして保存するファイル（ファイル名は mainData.js）が CSV ファイルからアプリ内で使用する JavaScript に変換されたファイルである。このファイルは Pamphwag で生成される index.html ファイルと同じディレクトリに配置されなければならない。

(3) 一覧表画面の設定

一覧表画面は見出し、一覧表、絞り込みボタン等から構成されるが、ここでは一覧表の部分の設定を行う。一覧表は多くの場合、表の要素（table 要素）として記述されると思われるが、必ずしも table 要素で記述しなければならない訳ではない。例えば箇条書き（ul 要素）で記述しても構わない。ただし、table 要素や ul 要素のように子要素（table 要素なら行の要素（tr 要素）、ul 要素なら li 要素）を含む要素でなければならない。この子要素が店舗数だけアプリ内で生成されて店舗の情報を一覧として表示することになる。従って店舗数だけ子要素が繰り返されることになるが、HTML の記述は子要素の繰り返しの前に親要素（table 要素等）の開始タグが必要であり、子要素の繰り返しの後に親要素の終了タグの必要である。その為、繰り返しの前のテキストエリアには table 要素等の開始タグを記述し、繰り返しのテキストエリアには子要素の HTML を記述し、繰り返しの後のテキストエリアには table 要素等の終了タグを記述する。例えばリスト 1 に示すデータの内、店舗名称と住所、カテゴリを一覧表に table 要素を用いて表示するには次のように各々記述する。

繰り返しの前

```
<table><tbody>
```

繰り返し

```
<tr><td class="shopname">店舗名称</td><td class="shopaddr">住所</td><td class="shopcategory">カテゴリ</td></tr>
```

繰り返しの後

```
</tbody></table>
```

繰り返しの部分ではデータを埋め込みたい要素には必ず class 属性を記述する必要がある。その class 属性値と店舗データの列番号の対応を指定することで、要素の内容が指定された列番号の値に置き換えられて表示されるようになる。（要素の内容ではなく指定された属性値に置き換えることも可能である。これにより、例えば img 要素の src 属性の値を置き換えることで画像を表示させることも可能である。）

class 属性値と店舗データの列番号の対応は、繰り返しの後のテキストエリアの下の表形式の入力欄に必要な項目を入力することで指定する。要素の class の欄には繰り返しの部分のデータを埋め込みたい要素に指定した class 属性値を入力する。列番号の欄には店舗データの列番号を入力する。種類の選択肢は置き換える対象を選択する。要素の内容を指定するとその内容が置き換えられる。属性値を指定すると属性値の欄で指定した属性の値が置き換えられる。列番号で指定されたデータに置き換える際には、前置きの欄に入力した内容がデータの前に、後置きに入力した内容がデータの後ろに追加されて置き換えられる。これにより、例えば店舗データの画像欄に画像のファイル名のみが記述されていて、画像ファイルが、/images ディレクトリ上に配置されていても、img 要素の src 属性の値を置き換えるには、種類を属性値、属性を src、前置きとして、/images/ を指定しておくとお適切に置き換えられる。（前置きを使用する具体例は次の情報画面の設定で示す。）表形式の入力欄の行数が足りない場合は「追加」ボタンをクリックすると 1 行追加される。余る場合は要素の class 欄か列番号欄を空白にすると、その行以降は無視される。これらの入力が完了したら「一覧表の設定」ボタンをクリックする。リスト 1 に示すデータの例を図 5 に示す。

要素のclass	列番号	種類	属性	前置き	後置き
shopname	2	要素の内容 ▾			
shopaddr	4	要素の内容 ▾			
shopcategory	6	要素の内容 ▾			追加

Fig. 5 Setting Example of Correspondence between Class and Column Number

(4) 情報画面の設定

情報画面は見出し、店舗情報の部分及び「前へ」「次へ」「戻る」ボタン等から構成されるが、ここでは店舗情報の部分の HTML をテキストエリアに記述する。HTML は全体が一つの要素の子孫要素となるように記述する。情報画面は指定された店舗の店舗データを表示する画面なので、データを埋め込みたい要素を明示し、その要素と店舗データの列の対応を指定する必要がある。そのため、埋め込みたい要素には id 属性を記述する必要がある。リスト 1 に示すデータなら次のような記述が考えられる。

```
<div>
<h2 id="infoName">店舗名称</h2>

<ul>
<li id="infoCate">カテゴリ</li>
<li id="infoAddr">住所</li>
<li id="infoGift">特典有無</li>
</ul>
</div>
```

データを埋め込みたい要素と店舗データの列番号の対応は、テキストエリアの下の表形式の入力欄に必要な項目を入力することで指定する。この表は要素の ID 欄に要素の id 属性値を指定する以外は一覧表の場合と同様である。これらの入力が完了したら「情報画面の設定」ボタンをクリックする。リスト 1 に示すデータの例を図 6 に示す。

要素のID	列番号	種類	属性	前置き	後置き
infoName	2	要素の内容			
infoImage	5	属性値	src	/images/	
infoCate	6	要素の内容		カテゴリ:	
infoAddr	4	要素の内容		住所:	
infoGift	7	要素の内容		特典の有無:	

Fig. 6 Setting Example of Correspondence between Id and Column Number

(5) 地図画面の設定

ここでは地図画面の有無、有する場合は地図の種類と URL を指定する。地図の種類は Google マイマップと OpenStreetMap に対応している。地図画面を表示させるには予めどちらかのサービスで地図を作成しておく必要がある。また、店舗データ中に店舗の緯度と経度の情報が含まれている必要がある。URL の欄には作成した地図の URL を入力する。緯度経度の列番号の欄には緯度の列番号と経度の列番号をカンマ (,) で繋いで入力する。最初の表示倍率は Google マイマップなら $z=16$ 等、OpenStreetMap なら $zoom=16$ 等と入力する。入力が完了したら「地図画面の設定」ボタンをクリックする。

(6) 並び替え機能の設定

一覧表は店舗データの何れかの列をキーとして並び替えることが出来る。一覧表画面で「並び替え」ボタンをクリックすると並び替えのキーのメニューが表示されるが、このメニューに表示されるキーの名称をソートキーの名称の欄に入力し、その列番号を列番号の欄に入力する。例えばリスト 1 に示したデータにおいて、店舗名称で並び替えたい場合はソートキーの名称には店舗名称と入力し、列番号には 3 と入力する（本来の店舗名称の列番号は 2 だが、並び替えはその読みで行うため 3 となる）。更に他のキーでも並び替えたい場合には「追加」ボタンをクリックすると 1 行追加されるので同様に入力する。入力が完了したら「並び替え機能の設定」ボタンをクリックする。

(7) 絞り込み機能の設定

店舗名での絞り込みを実装するには店舗データ中に店舗名の読みが必要である。実装する場合は読みの列番号を入力

する。

住所での絞り込みは町名等のチェックボックスにチェックを付けた町名で始まる住所の店舗のみに絞り込む機能である。町名等のチェックボックスは五十音順に並んでいるべきなので、店舗データに含まれるデータから生成することは困難である。その為、これを実装するには町名等の一覧表 (CSV ファイル) が必要である。

キーワードでの絞り込みは「マッチング方法をどれか含む」「全て含む」「切り替え可能」から選択し、マッチングの対象となるデータの列番号を指定する。お勧めキーワードがある場合には、お勧めキーワードをカンマ (,) で区切って入力する。

位置による絞り込みには緯度と経度の情報が必要であり、店舗データ中の緯度の列番号と経度の列番号をカンマ (,) で区切って入力する。

その他の絞り込みは「選択肢型の絞り込み」と「○×型の絞り込み」の2種類が実装できる。その他の絞り込みを実装するには名称を入力して種類 (選択肢か○×) を選択し、「設定」ボタンをクリックする。種類で選択肢を指定した場合、部品の種類と列番号、選択肢とその値の表を入力する。部品の種類はチェックボックス、ラジオボタン、選択ボックスから選択する。選択肢と値の表は、選択肢の欄にはラベルとなる文字列を入力し、値にはマッチングに使用する値を入力する。例えばリスト1に示したデータでは、カテゴリという名称で選択肢型の絞り込みが考えられる。部品の種類はチェックボックスが適切だろう。列番号は6である。選択肢と値はどちらも同じ文字列で、小売店、飲食店、ホテルの3つが考えられる。

種類で○×を指定した場合、選択肢、値、列番号の欄から成る入力欄の表 (最初は1行のみ) が表示される。例えばリスト1に示したデータでは、選択肢の欄には特典の有無と入力し、値の欄には○、列番号の欄には7と入力すると、特典有無の値が○の店舗だけに絞り込む機能を実装できる。

更にその他の絞り込みを実装するには「絞り込みの追加」ボタンをクリックし、新たに表示された入力欄に同様に入力していく。これらの設定が終わったら「絞り込み機能の設定」ボタンをクリックする。

(8) パンフレット・アプリの生成

(1)~(7)設定が終了したら、「パンフレットアプリの生成」ボタンをクリックする。すると PAG.js と index.html という2つのリンクが表示されるので、これをクリックしてファイルを保存する。更に、基本的な CSS ファイルや Pamphwag で生成された全てのアプリで共通に必要な JavaScript ファイル等を ZIP でまとめたファイルがあるので、これもダウンロードして解凍する。index.html と PAG.js、(2)で得られる mainData.js 及び ZIP ファイルを解凍して得られる main.css、PAfunc.js、ui_img フォルダは同じディレクトリに配置する必要がある。

以上が Pamphwag によるパンフレット・アプリの生成手順だが、この内の(3)及び(4)では HTML の記述が必要である。また、生成されたパンフレット・アプリはそのまま公開できるような実用的なデザインにはなっていないので index.html や CSS ファイル等を編集して、実用的なデザインのアプリケーションに仕上げる必要がある。そのため、HTML と CSS に関する知識は必須である。実用的なデザインに仕上げたら、ウェブサーバの適切なディレクトリ上に前述のファイルを配置することでパンフレット・アプリを公開することが出来る。なお、ウェブサーバは別途用意する必要がある。例えば、後述の「くるめ健康づくり応援店」アプリは、久留米工業大学にサーバマシンを用意して、kenko-ouen.kurume-it.ac.jp というドメインを取得し、https://kenko-ouen.kurume-it.ac.jp として公開している。

なお、Pamphwag では様々な情報を入力する必要があるが、入力された内容がサーバに送信されることはない。そのため、入力内容を再利用するためにはバックアップファイルを保存しておく必要がある。Pamphwag では JSON 形式のファイルとしてバックアップファイルを出力できるようにしている。

5. 使用評価

完成した Pamphwag を用いて、「くるめ健康づくり応援店」アプリの応援店の閲覧機能 (「くるめ健康づくり応援店」アプリは健康づくり応援店の閲覧機能と食事と体重の管理機能から構成されている⁽³⁾) の生成を行った。事前にバックアップファイル出力等の一部機能が完成していない時点で「久留米市地産地消推進店」アプリと「くるめ健康づくり応援店」アプリの応援店の閲覧機能の生成を行い、バグ出しとその修正を行っていた⁽⁴⁾ため、パンフレット・アプリは問題なく生成出来た。

生成された index.html や CSS ファイルを編集して、実用的なデザインとなるよう作業を行っていたが、データ項目

に変更が生じたため再度生成し直すことになった。新たに生成された index.html にはそれまで行っていたデザインの修正が反映されていないため、編集中だった index.html から修正箇所を新たな index.html へコピーする等の作業が必要だった。

またパンフレット・アプリ以外への応用として、「くるめ健康づくり応援店」アプリの食事と体重の管理機能における食事の記録機能の生成を試みた。食事の記録は、各食事（朝食、昼食、夕食、その他）に選択した料理のエネルギーや栄養素等が追加される形で記録するものであるが、その操作は料理の一覧表から料理を選択してその内容を確認して追加するようにしている。ここで、料理の一覧表やその絞り込み設定、料理の内容の確認画面（追加する画面）は追加する機能を除いてパンフレット・アプリと同様であるため、料理のデータさえあれば Pamphwag で生成可能であると考えられた。料理のデータは久留米市よりオープンデータとして CSV 形式のデータが公開されているので、そのデータを用いて Pamphwag で生成した。これも問題なく生成出来たので、生成されたファイルにプログラムを追加する等して食事の記録機能は実現出来た。このようにデータさえあればパンフレット・アプリ以外にも Pamphwag は利用出来る可能性がある。

6. まとめ

プログラミングのスキルを全く必要とせずに、パンフレット等のデータから容易にパンフレット・アプリを作成できるようにすることを目的として、パンフレット・ウェブ・アプリケーション生成ツール Pamphwag を開発した。Pamphwag ではプログラミングを不要とする代わりに、生成されるパンフレット・アプリのユーザインターフェースの自由度はある程度制限されてしまう。この点はより応用範囲の広いアプリ生成ツールとするには検討が必要である。ところで、パンフレット・アプリで重要な機能としては条件を指定して店舗の一覧を絞り込む機能があるが、Pamphwag では店舗名での絞り込み、住所での絞り込み、キーワードでの絞り込み、位置による絞り込み、その他の絞り込み（選択肢型の絞り込みと○×型の絞り込み）の機能を実装出来る。実際に「くるめ健康づくり応援店」アプリの応援店の閲覧機能の生成に利用し、問題なく生成できた。しかし、再度生成するとデザインの変更のための修正が反映されないため、余分な作業が必要となるといった課題がある。

7. 今後の課題

Pamphwag ではプログラミングのスキルは必要としないが、HTML 及び CSS に関する知識は必須である。そのため誰でも簡単に利用できるツールであるとは言えない。より多くの人が利用できるようにするには、画面を HTML と CSS を直接編集して作成するのではなく、市販のウェブページ作成ソフトで作成されたファイルを読み込んで生成する機能が必要であると考えられる。そしてこのような機能があれば、データ項目に修正が生じた等の理由で再生成しても余分な作業は必要なくなり、前述の課題も解決される。Pamphwag は現状では誰でも利用できるツールとはなっていないが、この機能の追加によって、ウェブページ作成ソフトさえ使えれば、誰でも容易にパンフレット・アプリを作成できるツールにしたいと考えている。

Pamphwag で生成出来るアプリはパンフレット・アプリだけではなく、多数の中から1つを選んで詳細を表示するといった機能のアプリなら生成可能である。しかし、多数の中から探すことが主目的である場合、検索のための画面が最初に表示された方が利用し易いことも考えられる。また、選択された項目の種類等に応じて詳細を表示する画面を切り替えられるようになっていた方がより多くの応用が考えられる。このように Pamphwag をより応用範囲の広いツールにするためには画面数や画面遷移等に自由度を持たせる必要があると考える。アプリ作成者が全くプログラムを記述することなく、このような自由度を実現することも今後の課題である。

文 献

- (1) “久留米市地産地消推進店アプリ”, <http://www1.city.kurume.fukuoka.jp/chisan/iphone/index.html>
- (2) キヤノン IT ソリューションズ, “超高速開発/ローコード開発 Web Performer”, https://www.canon-its.co.jp/products/web_performer/
- (3) 山田貴裕, “くるめ健康づくり応援店アプリの開発”, 久留米工業大学研究報告, No. 40 (2018), pp. 152-160
- (4) 山田貴裕, “パンフレット・ウェブ・アプリケーション生成ツール Pamphwag”, 第3回 ADADA Japan 学術大会 (2017)