

時を〇〇するソフトウェアコンテスト: アイデアを説明する文書

2015年10月8日(木)

タイトル 時をコーディネートするソフトウェア ~ Spare Time Coordinator ~
チーム名 STC...
エントリー番号 4番

要旨

予定に追われる生活の中で不意に生まれる空き時間。この空き時間を有効利用出来れば私たちの生活はきっと有意義なものに生まれ変わるはず・・・と頭では分かっている、目的もなくいつもと同じように時間を過ごしてしまった経験は誰しもが持っているのではないだろうか。

そんな空き時間をコーディネートするのが『[時をコーディネートするソフトウェア~Spare Time Coordinator~](#)』である。ユーザは時計型 UI で直感的に空き時間を入力、位置情報をもとにしたアルゴリズムによって複数のコーディネート結果を得ることが出来る。ユーザの負担を軽減したことで手軽に利用出来、ユーザにワクワク出来る新しい刺激を与えることも可能となるソフトウェアを実装出来た。

本文

1. アイデアの背景

私たちは日ごろ予定に追われながら生活をしているが、不意に予定が変わったり、無くなったりして、思ってもみない空き時間が出来た経験を誰しもが持っている。

アサヒグループホールディングスの調査[*1]によると、私たちが「暇だと感じる瞬間」は「30分間以上の時間が空いた時」と回答した人が全体の26%と最も多く、若年層ほどその時間が縮まる傾向があることが分かっている。

また「外出先で、暇な時間30分間で何をするか」というアンケートで、「書店・コンビニを見つけて本を立ち読みする」や「その場所の周辺を散策・散歩する」といった、特定の目的を持たず時間を潰す行動が上位に入っている。アンケート結果の詳細は図1に示す。

しかし、初めての旅行先や久しぶりに訪れた慣れない街では、現在地周辺の情報が分からず空き時間を持て余してしまったり、いつもと同じような行動をしてしまったりといった状況になることが考えられる。このことから、『時間』と『位置』の情報をもとに、楽しく有効な空き時間のプランをコーディネートするソフトウェアを創りたいと考えた。

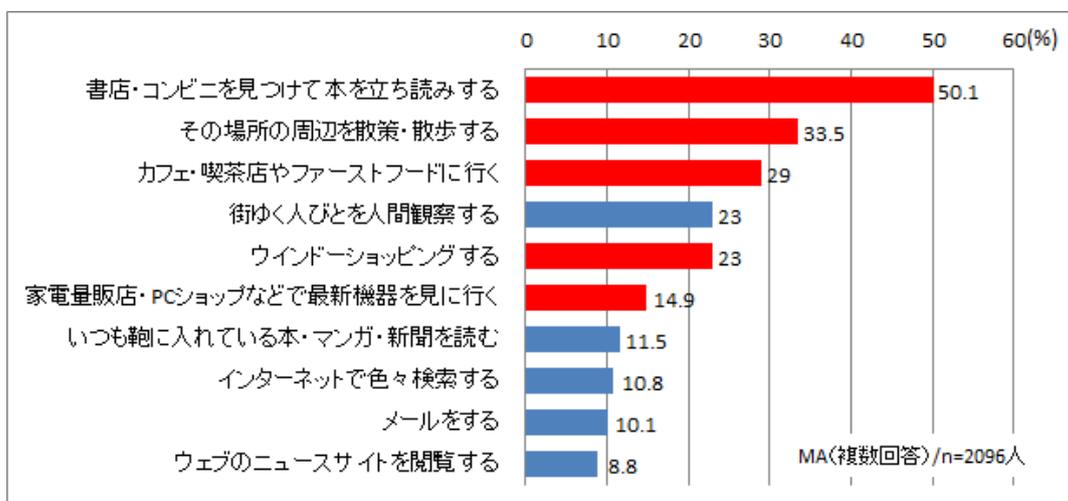


図1 <外出編>暇な時間30分で何をやる? [*1]アサヒグループホールディングス調査より

2. 提案ソフトウェアの目的

ユーザが提案ソフトウェアをいつもの場所で利用する場合には、いつもと違った空き時間のプランに出会う可能性が増える。一方で、初めての場所で利用する場合には、周辺情報が分からなくて空き時間を持て余すことがなくなる。

またユーザ側の負担を軽減するために、時計型 UI によって直感的に入力された時間情報と、自動取得される GPS に基づく位置情報のみで空き時間のプランのコーディネートを試みている。計算に用いる情報量が少ないことはあまいな結果に繋がりがうるが、提示するプランの質に強弱を設け複数の結果を表示することでユーザの幅広いニーズに応えることが可能となる。

将来的には、ユーザが行ったプランを保存し積み重ねていくことで、ユーザの手によってユーザのニーズを捉え続けるソフトウェアに成長させていくことが出来る。そうすればユーザは思いもかけないコーディネート結果に出会える事が出来、提供側はユーザ特性を他の分野に活用することが可能となる。

3. 時をコーディネートするソフトウェア

3-1 提案概要

時をコーディネートするソフトウェアは、『時間』と『位置』を基にプランをコーディネートする。

従来のレコメンドシステムがユーザにとって最適な結果を導くためには、ユーザ特性を学習するための事前情報が多く必要である。しかしユーザの入力情報が増え、利用時の負担を増やすとソフトウェアの利用しやすさが損なわれてしまう。そこで、提案ソフトウェアでは時計型 UI によって直感的に入力された時間情報と、自動取得される GPS に基づく位置情報のみからプランのコーディネートを行う。

また予定と予定の間に生まれた空き時間に手軽に提案ソフトウェアを利用することを想定し、「プランを実行するための準備・移動時間」を考慮しコーディネートを行う。これにより前後の予定を気にすることなくソフトウェアを利用することが出来るようになる。

プランの実行時間、目的地への移動距離によって難易度を設け、表示結果を見たうえで、ユーザが選択することが出来る余地を与えることとしている。これらの詳しいアルゴリズムは 3-2 節で示す。このように選択をしてもらうことで、少ない入力情報しかなくても、空き時間をのんびり過ごすプランや、アクティブに過ごすプランなど複数のプランを比較しながら、ユーザは自分に合ったプランを見つけることが出来るようになると考えている。

3-2 コーディネートアルゴリズム

ユーザの位置情報からユーザ周辺の情報を取得する際、全ての情報に対して検索を行うと計算量が膨大になるため、検索範囲を限定した上で、「プランを実行するための移動時間」に考慮した「プランを実行する時間」に基づきプランをコーディネートすることを行う。

図 2 は京都府を例にしたテーブル作成のイメージ図である。緯度と経度を用いて府全体を格子状に分割し、その格子ごとにプランを登録したテーブルを用意する。これによりプランのデータが増加しても検索時間が軽減可能である。

図 3 は各テーブルの範囲と、テーブル選択のイメージ図である。人の歩く速さを 80m/分として目的地までの往復時間を最大 10 分と考え、移動の最大距離は片道 400m と設定している。したがって登録するプランは、約 136m × 約 136m という非常に小さな位置情報の区画ごとにテーブルを分け、それぞれプランを位置情報に合わせて登録する。

またユーザの位置情報をもとにユーザが所属するテーブル(図 3 では(x, y)のテーブル)を判定、その周囲 8 つのテーブルを含めた合計 9 つのテーブルの中から、「プランを実行する時間」に応じた、時間に余裕があるのんびり過ごすプランや、時間ギリギリまでアクティブに過ごすプランを、プランの実行難易度とともに複数表示する。

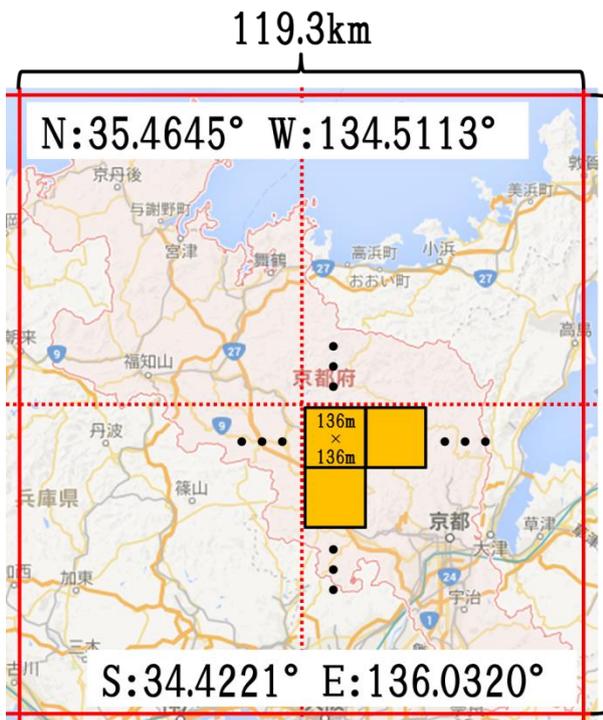


図2 テーブル作成

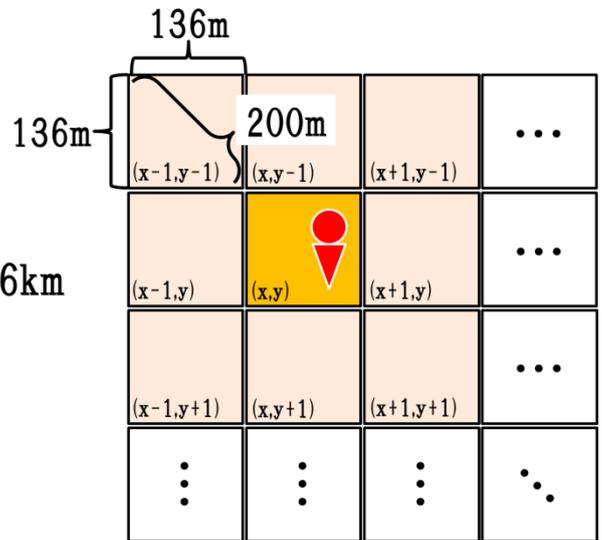


図3 テーブル選択のイメージ

3-3 利用例

提案ソフトウェアを利用する側は、3-2 節で述べたように移動時間にも考慮したプランをコーディネートするため、前後の予定に影響を与えず利用出来る。また知らない場所でも利用可能であり、時計型 UI は直感的で誰にでも使い易いものとなっている。例えば、映画や舞台を見る際に早く目的地周辺に到着した時、待ち合わせに友人が遅れ時間が出来た時、終電前のサラリーマンなど、いつでも、どこでも、誰でも利用出来る。

提案ソフトウェアを提供する側は、幅広いユーザに対しアプローチが可能であり多くのユーザ特性の収集出来る。また食事プランでは某グルメ検索サイトに移行、ウィンドウショッピングプランでは限定フェアを宣伝、各企業と連携しながらサービス提供出来る。また地域の町おこしイベントの情報など地域貢献に参画出来るなど利用可能性が広い。

現時点ではユーザから得る情報を「入力時間」と「位置情報」に限定しているため、ユーザにとって最適な結果を導くことが難しいケースが起こりうる。したがって、将来的にはユーザのスマートフォンにダウンロードされているスケジュールアプリや Twitter などの SNS アプリを利用してユーザの情報を得たり、ユーザの利用履歴からソフトウェアが学習したりということを行えばより適切なプランをコーディネートすることも可能である。

4. STC : Spare Time Coordinator

4-1 ソフトウェア概要

3 章で述べたソフトウェアのプロトタイプとして、STC の実装を行った。開発環境と実装内容を表 1、2 に示す。

ユーザが 4-2-1 項で示す時計型 UI で、空き時間をソフトウェアに入力すると、入力された時間とユーザの現在位置情報から、空き時間を楽しく過ごすことの出来るプランをソフトウェアがコーディネートする。コーディネート内容は飲食店のおススメや、近場の観光スポットなど幅広く用意する。またコーディネート結果は 4-2-2 項で示すように難易度順で表示しスクロールしながら確認することが出来る。システム構成ならびに、具体的な利用時のシステムの流れは、4-3 節で示す。

表 1 開発環境

項目	ソフトウェア
OS	iOS 5.0 ~
デバイス	iPhone5S ~
サーバ	Amazon EC2
DB	SQLite3

表 2 実装内容

項目	名前	言語	説明
STC	STC	Swift	時間の入力、結果表示
PlanSV	PlanSearch	PHP	プランを検索・決定
	geoSearch		ユーザの位置に基づく検索テーブルを決定
PlanDB (Table)	Places	SQLite3	プランの登録を行う作業テーブル 現在の使用可能エリアは京都市右京区嵐山周辺
	Place (1..n)		格子状のエリアごとのテーブル 「Places」からデータを振り分けられる
	Map		「Place」を特定するための位置情報を保持

4-2 実装における工夫点

4-2-1 時計型 UI

図 4 に示すような時間入力フォームや、コンボボックスなどによる時間の入力方法は良く知られているが、今回実装した STC では、直感的な時間入力 UI を採用している。その実装画面を図 5 に示す。

図 5 の左側の画面のように現在時間が表示されるメイン画面をタップして、長針を回すように指をスライドさせると、右側の画面のように時間を選択することが出来る。図 5 であれば 20 時 30 分過ぎの時刻から 21 時 10 分前までの 35 分間の空き時間があるということになる。また選択した時間は 5 分ごとに画面を塗りつぶすことで視覚的にも選択している時間が分かりやすいものとなっている。

時計型 UI を用いることで、選択した時間にどれくらいの幅があるのかということが直感的に分かりやすくなる。またユーザに新鮮さを与えるだけでなく、思わず針を回してしまうという面白さも含んでいる。



図 4 従来の時間入力フォーム

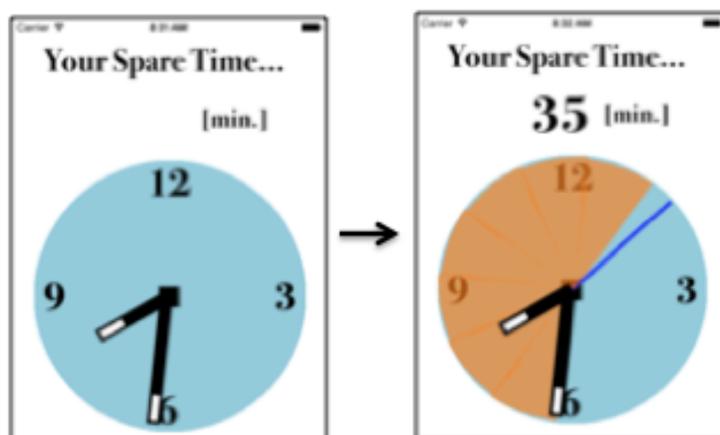


図 5 時計型 UI

4-2-2 難易度 GUI

入力時間に基づくプラン一覧の表示方法は、難易度順を採用した。難易度は☆マークを表示し視覚的に分かりやすくなるように工夫した。表示結果のイメージ図を図 6 に示す。またここでいう難易度は、ユーザが空き時間を過ごす際の行動の忙しさを示すものである。

入力された空き時間に対して、移動時間が必要となったりプラン自体を実行する時間がかかったり、プランの内容が充実した最もアクティブに過ごすプランを実行最高難易度 5、移動時間がほとんど必要なかったりプランを簡単に実行出来たりするプランを実行最低難易度 1 としてランク付けを行う。これらを実行難易度が高いプランから順に表示し、ユーザは表示されたプランを比較しながら、自分に合ったプランを選択することが可能になる。またプラン入力件数を増やしたり、将来的にユーザがプランを登録出来るようにしたりすれば、自分と同じように空き時間を過ごした人たちが実行した、多くのおススメのプランを比較しながら、時間の楽しみ方のバリエーションを感じる事が出来る。



図 6 プラン結果画面のイメージ

4-3 システム構成

STC のシステムは図 7 のように構成されている。利用時のシステムの流れに関しては、表 3 に示す。

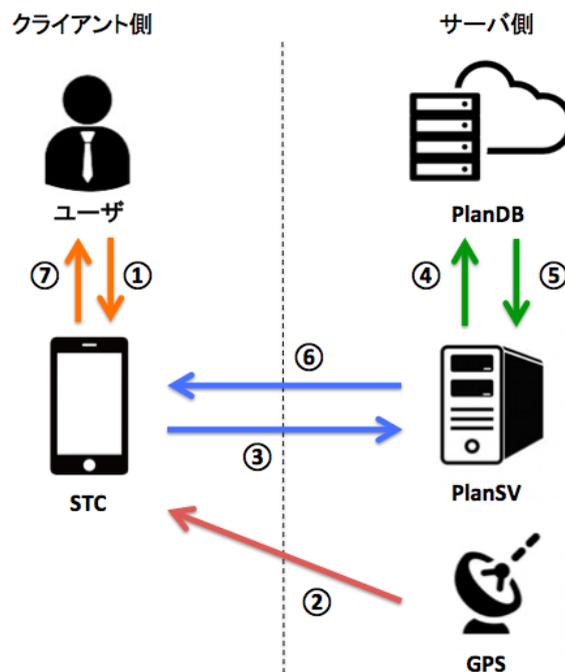


図 7 STC のシステム構成

表 3 システム構成の詳細と具体例

手順	内容	具体例 ユーザ: A さん 状況: 出張先(京都市内)で空き時間が 30 分ある
①	ユーザは、STC を起動し、現在時刻からの空き時間を入力する。	A さんは STC を起動し、空き時間を 30 分に指定する。
②	GPS は、ユーザが検索を実行した際に、ユーザの現在位置情報を STC に送信する。	GPS は A さんの現在位置(35.012876, 135.677741)を STC に送信する。
③	STC は、PlanSV に入力情報(ユーザの空き時間と現在位置情報)を送信する。	STC は、PlanSV に、A さんが現在位置(35.012876, 135.677741)において 30 分の空き時間があるという情報を送信する。
④	PlanSV は、③において受信した情報に適したデータ(プラン)を取得するため、PlanDB に対してクエリを投げる。	PlanSV は、(35.012876, 135.677741)において、30 分で行動出来るプランを取得するために、その条件をもとにしたクエリを PlanDB に投げる。
⑤	PlanDB は、④において要求された条件に合ったデータを PlanSV に送信する。	PlanDB は、④のクエリに適したプランとして、 1.天龍寺を訪れる 2.タリーズコーヒー嵐電嵐山駅店でゆっくりする 3.渡月橋周辺でウィンドウショッピングを楽しむなどの候補をデータとして PlanSV に送信する。
⑥	PlanSV は、⑤において PlanDB から受信したデータを STC に送信する。	PlanSV は、⑤において PlanDB から受信したデータを STC に送信する。
⑦	STC は、受信したデータを閲覧出来る状態にして、ユーザに対して、空き時間におけるプランの提案を行う。	STC は、受信したデータをプラン一覧表にして表示することで、A さんに対して、空き時間におけるプランを提示する。

5. 参考文献

[*1]: アサヒグループホールディングス、毎夕アンケート第 513 回暇な時間が出来たら何を？、2014 年 7 月、
URL【<http://www.asahigroup-holdings.com/company/research/hapiken/maian/201407/00513/>】