

Kadecot:スマートハウスのいじれる化プラットフォーム

大和田 茂* 徳久 文彬† 市岡 陽子 ‡ 島崎 聡§

概要. Kadecot(カデコ)は,家電ネットワークを利用するアプリをWebブラウザ内で制作するための開発環境である. これまでも様々な家電ネットワークサービスが提案されてきたが,ユーザーコミュニティによるサービスの創出やデザインにフォーカスしたシステムは少なかった. 本研究ではこの点に着目し,UGC的な力を家電サービス開発に生かすための試みとして,サービスロジック開発と画像や音声などのリソース制作を分離することができ,JavaScriptを独自拡張したスクリプティング言語システムによりそれらを統合しつつ機器制御を可能にするWebブラウザ内SDKを開発した. 本システムを用いることでプログラマは最小限のテキストメッセージと機器制御コマンド,そしてネット上のリソースのロードコマンドを記述するだけでリモコンアプリを作成することができる. 他方リソース供給側も使い道を具体的に特定せずに制作を行うことができる. 画像リソースは一枚絵のみならずボタン機能やアニメーションデータなどを付加することもでき,それらはJSONPでの応答が可能なコンテンツサーバの下に配置することで一般公開できる. これらの機能を組みあわせると,ボタンベースのリモコンからビジュアルノベルスタイルのゲームまで作成することが可能であり,擬人化やヒューマンエージェントを用いたエンタテインメント性の高いリモコンアプリも作成することができるようになっている.

1 背景

住宅業界では2012年はスマートハウス元年などと言われている. かねてより世界的に関心が高まりつつあった循環型社会や再生可能エネルギーといった概念が,こと日本においては東日本大震災を契機として現実の住宅や,それを構成する創電・蓄電・生活家電といったプロダクトとして急速に市場に登場してきている. 政府による積極的な支援策も功を奏して,2012年後半のスマートハウス市場はさながら戦国時代の様相を呈している.

しかし,実際には初期投資が大きくメンテナンスコストもかかることから一般家庭で導入する規模のシステムでは経済的メリットは生じにくいのが現状である. また,新たな価値の創出より無駄の削減に重心が置かれているために購入後の満足感が持続しづらいついて考えられている. そのため,現在の関心の高まりは一過性のブームで終わってしまう可能性もあり,ここに政府や各家電メーカーは危機感を抱いている.

実際にはエネルギーマネージメント以外でもネット家電のメリットは大きいと考えられ,研究の舞台では様々なサービスが提案されてきた. いくつかの商用システムでは実際にそれらを実装し,電子鍵や給湯器の遠隔操作や調理レシピのダウンロード,食

材管理などが可能なシステムを消費者の手の届くところに投入してきた. しかし,どれも商業的に大成功を収めたとは言えない. その原因は導入コストの他にもいろいろ考えられるが,結局家電ネットを応用した生活サービスは消費者にとっては今もって未開の地であり,何がどのような条件下で価値を持ち,どのような市場が生まれ得るのかなど,多くのことが手探り状態であると言える.

我々はこういった状況を打開する手がかりをUGC(User Generated Content:ユーザー生成コンテンツ)またはCGM(Consumer Generated Media:消費者生成メディア)と呼ばれる,ユーザー自身がサービスを創り出す文化に求めた. その理由は以下の通りである.

- 生活における価値観は様々であり,マスを意識して作られたアプリをそのまま使うだけでは細やかな要望に応えられない. しかし,アプリ自身をユーザーが自由に開発・改変できれば,それぞれの特殊な要求を充たすようなサービスを実現できる道が開ける.
- スマートハウス向けアプリは生活の中に組みこまれていくものであり,単機能の軽いアプリが適していると考えられる. このようなアプリは企業が大規模に開発するよりも個人の創造力で様々なアイデアを実装した方が多様性が増し,絶対数の増加も見込める.
- 複数のベンダーからなる機器間連携サービスでは責任の所在が曖昧になりやすいため,そもそも企業は積極的になりにくい. 他方,UGC

Copyright is held by the author(s).

* Shigeru Owada, 株式会社ソニーコンピュータサイエンス研究所

† Fumiaki Tokuhisa, 東京工業大学

‡ Yoko, Ichioka, お茶の水女子大学

§ Satoshi Shimazaki, 東京工業大学

により作られたアプリではもし問題が起ころとも個人対個人の関係として理解される傾向があり、新しいチャレンジに適している。

このような考えに基づき、Kadecot は家電ネットワークを操作するプログラミング環境を整備し公開することを目的として開発されている。我々はこれを、エネルギーの「見える化」にちなんで「いじれる化」と名付けている。

これまでも様々な観点から複数家電の協調動作設定を簡単にするためのハード面・ソフト面からの工夫はなされてきていた。しかしながら、それらは機能を連携動作設定のみに限っていたり、より汎用性が高いものであってもプログラミング環境が汎用目的なだけであって、様々な専門性や興味を持った開発者が力を合わせて開発を進められるような環境整備ははかられておらず、暗黙のうちに単独開発者による開発を前提としていた。本システムではとりわけ大きな専門性の違い、つまり、サービスロジックの開発と画像等のリソースの開発プロセスを分離し、独立して作業を進められるようにした。そして、Web 上に提供されたコンテンツサーバの出力を統合してアプリを完成させるための開発環境を提供している。

これらの機能を総合的に用いると、ボタンベースの統合リモコンアプリのみならずビジュアルノベルと呼ばれる 2D キャラやストーリーを楽しむタイプのゲームも作成することができるため、擬人化やヒューマンエージェントによるエンタテインメント性の高い対話型リモコンアプリなども容易に作成することができるようになっていく。

2 既存研究

住環境を情報化することにより実現するサービスの研究には比較的長い歴史がある。最も初期のシステムの一つとしては、1966 年に発表された ECHO IV がある [1]。1987 年までにはスマートハウスの概念が提唱されており [2]、その後 TRON 電脳住宅の実装や [3]、ユビキタスコンピューティング概念の萌芽 [4] がみられた。この時期に各種家電の遠隔操作や在宅ヘルスケアシステムの実用化提案などがなされたが [5]、広範な市場創出には至らなかった。2000 年代に入ると、通信インフラがブロードバンド化され、機器の遠隔操作や Web カメラの導入などネットサービスと連携したシステムが盛んに試作された [6]。近年では世界的な環境問題への関心の高まりと [7]、スマートグリッドの普及によって [8]、主にエネルギーの文脈からスマートハウスが語られている。日本国内では政府や業界団体主導でのスマートハウスプロトコルのインフラ整備も進められているが [9]、その上で動作する具体的なアプリケーションの一般的な認知やメーカーによるサービス開発は遅れている。

また、ネットワークで接続された家電を協調動作



図 1. 様々な環境で実行した例



図 2. メインメニュー

させるための装置 [10] やルール定義システム [11]、リコメンデーションシステム [12] などの研究提案も多々存在するが、広く使われるには至っていない。

UGC や CGM と呼ばれる、ユーザー自身がコンテンツを生成する仕組みは特筆すべき近年の潮流である [13]。同人ゲームはもちろん [14, 15]、大規模な辞典の創出や [16]、プレイヤー自身の創作によって多様な遊び方を許すようなゲームの登場 [17]、特定キャラクターマーケットの自己成長 [18] など枚挙に暇がない。こういったシステムはブームを乗り越えて現在では完全に市民権を得ているが、今後一層広い分野に適用されていくものと思われる。

3 提案システム

Kadecot は HTML5 で開発されたアプリケーションであり、ほとんどの Web ブラウザ内で実行することができる (図 1)。Web ブラウザのサンドボックス内からホームサーバやコンテンツサーバに接続するには JSONP, WebSocket, または (Android アプリ版に限り) Java オブジェクトに直接アクセスする方法 (addJavascriptInterface) を用いている。Android アプリ版はライブ壁紙としても動作することができる (図 1 右)。

3.1 ユーザーインターフェース

Kadecot のトップメニューには「アプリ一覧」と「機器ネットワーク」の二つのタブがある (図 2)。「アプリ一覧」(図 2 左) は、Web 上に存在する Kadecot アプリへのブックマークが並んでおり、どれかを選択すれば Web からアプリのリソースが読みこまれて実行がはじまる (ここで言う「アプリ」とは Kadecot

Kadecot:スマートハウスのいじれる化プラットフォーム



図 3. アプリの設定画面からの SDK ツール起動

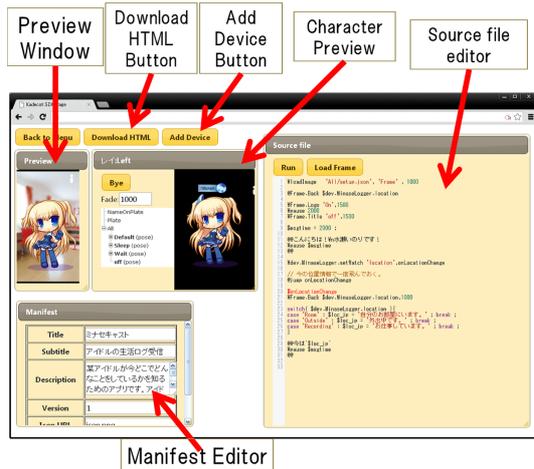


図 4. SDK の画面構成

そのものを意味するのではなく、Kadecot というプラットフォーム上で動作する一つのコンテンツという意味である。「機器ネットワーク」のタブを選択すると、現在利用可能なデバイス(家電オブジェクト)とプロトコルの一覧が表示される(図2右)。初期状態では、デバイス一覧に機器のソフトウェアエミュレータが含まれており(現状では電力ロガーとエアコン、ブルーレイレコーダー、位置センサー)、プロトコル一覧には DLNA と住宅 API, 赤外線ターミナルが含まれている。実際の家電機器をデバイス一覧に加えるには、「プロトコル」の中から適切なものを選び、現れる設定画面の中で「機器発見」を行えばよい。各デバイスインスタンスにはユニークな「ニックネーム」がつけられており、全く同じ型番の機器が複数存在していても区別して扱うことができる。

3.2 サービスロジックの開発方法

プログラミングを開始するには、アプリリストの一番下の Add Bookmark から新規のアプリ作成を開始するか、または自分の作りたいものに近い既存アプリがあれば、そのアプリの設定画面の中にある「ソースをエディットする」というボタンを押すことで、そのアプリのソースコードが表示され、それを変更することができる(図3)。

図4が SDK の外観である。プログラマーがエディットするのは Source File ウィンドウの中に表示されているプログラム本体、およびアプリのプロパティ情報を含むマニフェストファイルである。マニフェストファイルは図4左下の Manifest Editor 内の項目をエディットすることで変更することができる。

作成中のアプリを実行するには Source File ウィンドウ内左上にある Run ボタンを押す。Preview ウィンドウ内でコンテンツが実行される。

SDK は広大な画面を必要とするので、実質的に PC 上のブラウザで使うことを想定している。この場合は通常の JavaScript(以下 JS と表記)のデバッグに用いられるメニューを出したり FireBug などのアドオンを導入することによって開発効率を向上させることができる。また、Download HTML ボタンを押すと JS に変換された後のソースファイルがダウンロードされてローカルにセーブされるので、これを実行しながらバグを特定することも可能である。このダウンロードされた html ファイルはアプリの配布に用いることもできる。

3.2.1 ソースコードの文法と SDK による支援

Kadecot のプログラマーは基本的に JS を用いてプログラミングすることになるが、(他の言語と同様に)JS の文法には独特な部分があり初心者には学習が必要であることから、JS をベースにしつつも最小限の機能に関しては簡潔に記述できるような拡張表記を導入することにした。拡張表記は C 言語におけるマクロのようなもので、プリプロセスにより全体が JS のコードに変換されてから実行される。

拡張表記部分はメッセージ表示、リソースやハードウェアの制御、基本的な分岐、ボタン応答など限られた機能しか実現していないが、これまでのシステムでは難しかったこと、つまり Web 上のリソースを組み合わせて見た目の質を維持しながらアプリロジックに集中するような開発を容易にしている。もちろん拡張表記を用いずに通常の JS だけで記述することもできるので、スキルのあるプログラマーであれば直接高度な処理も書くことができる(もちろん Kadecot のフレームワークに関する理解は必要である)。紙面の都合からこれらの拡張文法の詳細について説明することはできないので、ここでは機能の一部についてユーザーが記述するソースと期待される動作のみ記すことにする。

まず、プログラムの最初で画面全体を定義する「フレーム」を読みこまなければならない(図5)。Source file ウィンドウ(図4右)内の Load Frame ボタンを押すとフレーム選択ダイアログが開き、Web 上のリソース URL を入力することができる。プリセットデータを用いる場合は選択肢から選ぶとそのデータが存在する URL がセットされるので、ここではプリセットから「Basic」を選ぶことにする。Load ボタ



図 5. フレームの挿入

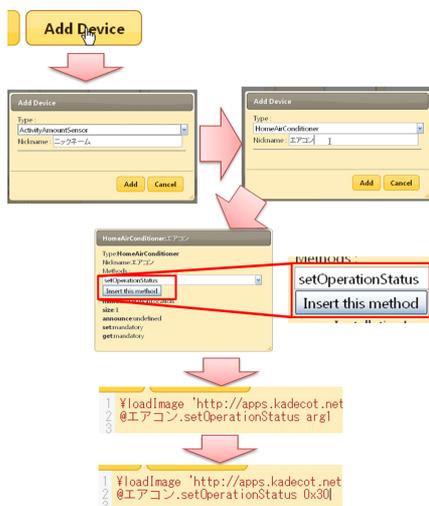


図 6. デバイスオブジェクトの挿入

ンを押すと Source file ウィンドウの中にフレームを読みこむコードが挿入され、同時に挿入されたフレームデータのプレビューと内部構造を示すウィンドウが開く(図5下)。

次に、エアコンを ON するコマンドを入力する(図6)。これにはまずエアコンデバイスの使用を宣言しておく必要があるので、画面左上にある Add Device ボタンを押す(図4上部)。すると Add Device ダイアログが開き、デバイスの種類とニックネームを指定するようになる。このニックネームの指定は、ソースの中でこのデバイス指定するときの ID になるので重要である。Type として HomeAirConditioner を選び、Nickname は「エアコン」として Add ボタンを押す。すると、エアコンの持つメソッドを選択するボックスを含むデバイスウィンドウが開く。所望のメソッド(ここでは setOperationStatus)を選んだ上で Insert this method ボタンをクリックすると、電源 ON 用のコマンドがソースファイルに挿入される。ただし、メソッドの引数は自分で書き換えなければならないので、ここでは電源 ON を意味す



図 7. サンプルの実行例

る 0x31 を引数として与える。

できあがるソースは二行で、以下の通りである。

```
\loadImage 'http:...', 'Frame', 1000
@エアコン.setOperationStatus 0x30
```

次に、電源 ON と OFF をするボタンを持ったアプリを作成する。これには select 文を用いた分岐を用いる。select 文は GUI によるサポートはないが単純な文法である。以下の例では、テキストメッセージの表示(@@からはじまる行)を行ったのちに ON/OFF をボタンで選ばせるようにしている。

```
\loadImage 'http:...', 'Frame', 1000
@@電源の ON か OFF を選んでください。
\select 'ON', 'OFF'

#ON
@エアコン.setOperationStatus 0x30
@@ON にしました！

#OFF
@エアコン.setOperationStatus 0x31
@@OFF にしました！
```

このソースを実行すると、図7左のようになる。

なお、#からはじまる行はラベル行であり重複は許されないが、select 行では選択肢の文字列を“オンにする:ON”のように書くことで表示する文字列とジャンプ先のラベル名を別にもできるので大きな不都合はない。

フレーム内に「サブスクリーン」というものが定義されていれば、外部のリソースサーバから提供される画像やセルアニメーション機能を持ったキャラクタ画像などをそこに読みこむことができる。例えば EduPack というフレームには4つのサブスクリーンが存在し、それぞれ別の外部画像を埋めこむことができる。ここにプリセットのキャラクタ画像を読みこんでセルアニメーションさせるコードは以下のようになり、実行結果は図7右になる。

```

\loadImage 'http...EduPack.json'
                                , 'Frame', 1000
\loadImage 'http...Char.json'
                                , 'フミヒサ', 1000, 'Left'
@フミヒサ.LArm 'Up', 1000
@フミヒサ.Mouth 'Laugh', 1000

```

一行目でフレームを読みこみ、二行目でフレーム内の'Left'という名前のサブスクリーンにキャラ画像を読みこんでいる。

※実際には引数の切れ目であっても拡張文法中での改行は許されていない。

3.3 リソースの開発・公開方法

画像や音声などのリソースはリソースサーバから提供される。ほとんどのリソースは単にHTTPサーバからの直接のファイルアクセスによって公開されればよいが、画像リソースに対してポーズ入れ換えのためのセルアニメーションやクリックされた際のコールバックなどの設定を行うには独自の設定ファイルを用意する必要がある。この設定ファイルはjson形式のファイルになり、これをWebブラウザのクロスドメイン制限を越えて利用できるようにするためにはKadecot本体からのJSONPリクエストに応答できるサーバでなければならない(通常はcgiプログラムによりJSONPに응答する。我々はそのためにphpで開発されたメディアプロバイダcgiも提供している)

この設定ファイルは複数の画像を階層的に管理し、かつ適切な位置に表示するための情報を持っている。一つの階層に含まれる画像群はデフォルトでは同時に重ねて表示されるが、ポーズの入れ換えを行うためにラベルを設定し、一つのみをセレクトして表示することも可能である。さらに、タイミングを指定して画像切り換えを行うことでパタパタアニメーションを行ったり、画像がタップされた時にコールバックするための機能、さらに他の画像を読みこむためのサブスクリーン設定なども行うことができる。

このファイルは単純なjsonファイルであり、テキストエディタで作成することもできるし、専用のPhotoShopプラグインも存在するのでこれを用いて自動生成することもできる。PhotoShopプラグインを用いる場合、各画像(レイヤー)に付随する情報は、PhotoShop内ではレイヤー名の一部として記述することになる(図8)。細かい文法の説明は紙面の都合上省略するが、ポイントは本フォーマットでは家電操作やプログラミングなどの知識が全くなくても様々な機能を持った画像リソースを作成でき、リソースサーバから公開することで種々のアプリ内で利用される可能性があるという点である。

```

1  {
2  , "version": "120329"
3  , "width": 227
4  , "height": 360
5  , "type": "stacklayers"
6  , "contents":
7  [
8  {
9    , "type": "switchlayers"
10   , "name": "Question"
11   , "cid": -1
12   , "key": "Question"
13   , "contents":
14   [
15     {
16       , "type": "image"
17       , "x": 0
18       , "y": 0
19       , "url": "Question.Question.png"
20       , "name": "On"
21       , "value": "On"
22     }
23   ]
24   }
25 ]

```



図 8. 画像設定用 json ファイル(上)と、元となる PhotoShop のレイヤー名と構造(下)

4 アプリの例

本システムを用いれば様々な機器連携アプリを作成することが可能である。

図9上は「家の中のもう一つの家族」というコンセプトで制作している「萌家電」というアプリで、各家電が擬人化して登場し、人間にサービスを与えるために協調して様々な活動をするというものである。例えば映像コンテンツを視聴する時にはライトを暗くしたりカーテンを閉じたりする、あるいは、家電を組み合わせると節電を達成するとその家電キャラ同士のストーリーが進展する、などである。

図9左下は「Starnight」で、電力ロガーの情報から住宅の節電度を計り、その変化を画像とBGMで知らせるためのアプリである。このアプリにはキャラクターもテキスト表示もなく、起動すると自動的に表示が変化していく。StarNightはAndroid版Kadecotのライブ壁紙として実行することを想定しており、既存の「見える化」アプリとは異なった、デザイン性の高いものを目指している。

図9右下は、エアコンと連動して動作する「サーモ」というアプリである。このアプリは、背景に描かれたキャラクターが部屋の温度についての好き嫌いを発言する。キャラクターの好きな温度を長時間持続させると好感度が上がり、衣装が増えるようになっている。この「好きな温度」は、デフォルトでは

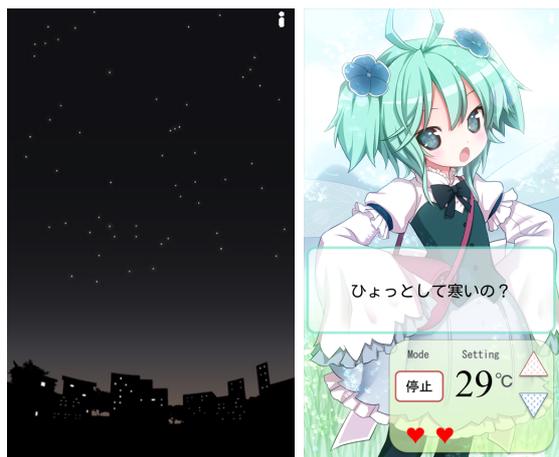


図 9. アプリ例:上は「萌家電」, 左下は「StarNight」, 右下は「サーモ」

28 度に設定されているが、ソースコードの先頭にこの値をリターンする関数が含まれており、これを改変すると動作が変化する。「いじれる化」の入門編として使われることを期待している。

5 まとめと今後の課題

Kadecot はスマートハウスの「いじれる化」を実現するプラットフォームとなることを目指したアプリ開発環境である。サービスロジックの開発とリソース制作は分離して進めることができるため、UGC コミュニティによるマッシュアップを通じて新たなサービスの開発を期待することができる。

本システムはハウスメーカーや携帯端末メーカー等、他社とのジョイントイベントや、Android アプリケーションの公開を通じて積極的にコンセプト・実装の宣伝公開を行っている。今後は開発システムを使ってもらうワークショップなどのイベントを企画したいと考えている。また、ユーザー自身が作成したプログラムをソーシャルな環境を通じて流通させる仕組みも開発したい。アプリの流通にはセキュリティの問題が非常に重要なので、安全を保証する仕組みも必要である。また、サポートするプロトコルとして、近年成長著しく、また家電のみならずセンサーや蓄電池、給湯器や電動シャッターなども規定され

ている ECHONET Lite 対応デバイスを積極的にサポートしていくことも挙げられる。

参考文献

- [1] If You Can't Stand the Coding, Stay Out of the Kitchen: Three Chapters in the Hi. <http://www.drdoobbs.com/if-you-cant-stand-the-coding-stay-out-of/184404040>, 1966.
- [2] R.L. Smith. *Smart House: The Coming Revolution in Housing*. Gp Courseware, 1987.
- [3] 坂村健. 電脳住宅・電脳都市-生活環境におけるコンピュータ化のあり方(人を対象としたシステム)-(人を対象としたシステム-いくつかの事例). 電気学会論文誌 D 産業応用部門誌, 111(5), pp. 347-348, 1991.
- [4] M.Weiser. *The Computer for the 21st Century*. Scientific American Special Issue on Communications, Computers, and Networks, 1991.
- [5] 郵政 I H 研究会. インテリジェントハウス読本 暮らしと情報の明日をひらく. ビジネス社, 1988.
- [6] <http://www.eclipse-jp.com/jeita/>.
- [7] A. Gore. *An Inconvenient Truth. The Crisis Of Global Warming*. Bloomsbury, 2007.
- [8] *American Recovery and Reinvestment Act of 2009*. U.S. Congress, 2009.
- [9] <http://www.echonet.gr.jp/>.
- [10] Kentaro Fukuchi, Takeo Igarashi, Maki Sugimoto, Charith Fernando, and Masahiko Inami. Push-pins: Design-by-user approach to home automation programming. In *International Conference on Ubiquitous Computing, workshop 5: ArchiBots 2009, ubicomp 2009*, 2009.
- [11] Kouji Nishigaki, Keiichi Yasumoto, Naoki Shibata, Minoru Ito, and Teruo Higashino. Framework and rule-based language for facilitating context-aware computing using information appliances. *2012 32nd International Conference on Distributed Computing Systems Workshops*, Vol. 3, pp. 345-351, 2005.
- [12] 川原圭博, 司化, 猪鹿倉知広, 登内敏夫, 森川博之, 青山友紀. 行動履歴と制約条件を考慮した情報家電制御機構. 情報処理学会研究報告. MBL, [モバイルコンピューティングとユビキタス通信研究会研究報告], Vol. 2006, No. 14, pp. 55-60, 2006.
- [13] 伊知地晋一. CGM マーケティング 消費者集合体を味方にする技術. ソフトバンククリエイティブ, 2006.
- [14] <http://www.nscripter.com/>.
- [15] <http://kikyuu.info/tvp/>.
- [16] <http://ja.wikipedia.org/>.
- [17] <http://www.japan.ea.com/spore/>.
- [18] <http://www.crypton.co.jp/cv01>.