

ハードウェア情報に基づく動的適応型ユーザインタフェース

Dynamically Adaptive User Interfaces based on Hardware Information

白須 光†
Akira Shirasu

新井 浩志†
Hiroschi Arai

1. はじめに

近年端末の種類が多様化しており、多くのユーザは場所や目的によって PC や携帯電話などの端末を使い分けている。このような多様な端末環境においても同一のソフトウェア（以下 SW）を同じようなユーザインタフェース（以下 UI）で利用できれば、ユーザの利便性が向上する。しかし SW の開発者は端末毎にそのハードウェア（以下 HW）に適した SW を開発する必要がある。特に携帯電話などの小型モバイル端末では、描画領域を有効活用し、入力装置に適した使いやすい UI を端末毎に作り変えなければならない。本提案では端末の HW 情報に基づいて UI を動的に適応させることにより、HW 環境を意識しない SW 開発・実行環境を提供するための一手法を提案する。

2. システム概要

本手法では HW プロファイルと UI 構成情報に基づいて、端末の HW に適した UI を生成する。HW プロファイルでは端末の画面サイズや解像度などの情報を記述する。UI 構成情報では UI の表現に必要なメニュー階層やウィジェットを記述する。レンダリングエンジンは HW プロファイルと UI 構成情報をもとにして実際の UI を構築し、描画する。UI 構成情報とレンダリングエンジンについては次節以降で詳しく説明する。

近年 UI を端末 HW に適応させるためのいくつかの研究が報告されている。Dygimes^[1]ではタスクの仕様を ConcurTaskTree (以下 CTT) という状態遷移木で定義している。CTT の 1 つのノードはタスクの或る状態における UI 構成情報を表し、表示するウィジェットとインタラクションで定義される。ウィジェットの位置関係は、left-of, right-of, above, below の 4 通りで相対的に指定する。レンダリングエンジンは HW プロファイルで定義された画面サイズに応じて極力この相対的位置関係を満足させるようにレイアウトを決定する。ウィジェット間の位置関係の指定に矛盾が生じた時は、優先度の高いウィジェットの位置関係指定を優先する。Dygimes はこの方法によってある程度画面サイズの変化に対応できる。しかし、近年の大画面 PC から携帯電話のような極端に小さな画面をもつ端末まで幅広く適応させることは困難である。

またマルチメディアコンテンツに焦点を当てた研究^[2]では、端末の画面サイズと最大表示色数に応じてマルチメディアコンテンツの形式変換を行う手法を提案している。マルチメディアコンテンツ表示の設定権限を UI 構成情報に記述しておくことにより、ユーザの意志によるマルチメ

ディアコンテンツの表示/非表示を設定可能にしている。しかし基本的に表示するウィジェットの数は一定であり、様々な画面サイズに応じてメニュー等を適応させることはできない。

提案するフレームワークでは HW プロファイルをもとにメニューの表示方法を動的に変更することで UI を端末に適応させる。本提案で想定する UI の画面構成を図 1 に示す。画面はメニュースペースとワークスペースからなる。メニュースペースはメニューウィジェットを表示する領域であり、メニューバーエリアとメニューボタンエリアに分けられる。メニューバーエリアはプルダウンメニューをラベルで表示するエリアであり、そのアプリケーションの全ての機能にアクセス可能とする。メニューボタンエリアはアプリケーションでよく利用される機能を選択しやすいように、メニューバーエリアの機能項目から抽出したボタンを表示する。本手法では HW プロファイルに応じてメニュースペースを動的に適応させる。

ワークスペースはアプリケーション特有の主たる目的を果たすための領域である。ワークスペースについてはアプリケーションによる仮想画面への描画をサポートし、仮想画面から実際の端末への適応はズーム機能とスライドバーで対応する。

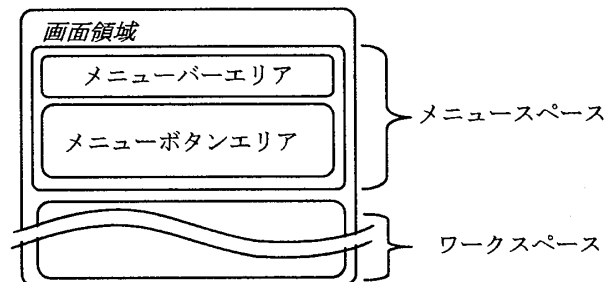


図 1. 提案する UI の画面構成

3. UI 構成情報

UI 構成情報ではメニューの構成をメニュー分類木の集合として定義する。メニュー分類木はメニュー分類項目や画面状態、呼出手続などを木構造で定義したものである。メニュー分類木の例を図 2 に示す。

1 つのメニュー分類木は多分木で、或るウィンドウ状態におけるメニュー階層を表わす。メニュー分類木のノードは、メニュー階層の根を表す状態ノード、メニュー分類項目を表す分類ノード、呼出す手続きを表す呼出手続ノードから構成される。中間ノードはすべて分類ノードである。葉ノードは手続きノードであるか、または他のメニュー分類木へのリンクを表す分類ノードとなる。これにより別のウィンドウへの遷移を記述することが可能である。ノードを繋ぐアークはメニューの繋がりを表す。

†千葉工業大学 工学研究科 電気電子情報工学専攻,
Master's Program in Electrical, Electronics and Computer
Engineering, Graduate School of Engineering, Chiba Institute
of Technology.

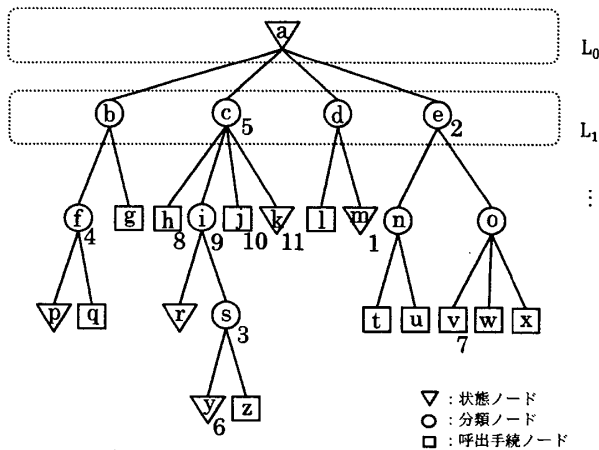


図2. メニュー分類木例

メニュー分類木のノードのうち、メニューボタンエリアに表示する項目には優先度 p を付与する。この値が小さいほどメニュー項目としての優先度が高い、すなわち選択される頻度が高いメニュー項目を表す。

4. レンダリングエンジン

レンダリングエンジンは HW プロファイルと UI 構成情報に基づいてメニュー階層とウィジェットのレイアウトを決定し、メニュースペースに UI を描画する。以下では画面領域に対して 1 行に表示可能なノード数の上限を N_{rmax} 、1 列に表示可能なノード数の上限を N_{cmax} とする。またメニュー分類木の根ノードからの深さを $\{0, 1, 2, \dots, n\}$ としたときに、階層 i に含まれるノードの集合を L_i 、メニュー階層 i の総ノード数を N_i 、ノード a の子ノードの数を N_a とする。

メニューバーエリアには、メニュー分類木の L_1 のノードを通常のプルダウンメニューとして表示する。複数のノードを 1 行または 1 列に表示する場合は、分類木において左側にあるノードを左または上に表示するものとする。画面領域が十分に確保できる場合には、 N_1 個の全ノードをメニュー項目として表示し、通常のプルダウン動作を行う。但し、画面領域が小さい場合には、メニュー分類木で指定されたプルダウンメニューの全てを表示できない場合がある。以下プルダウンメニューを表示できない 3 通りの場合について、メニューを画面に適応させるための方法を示す。

まず、画面の横幅がプルダウンメニューの初期状態の横幅より小さい場合、すなわち $N_{rmax} < N_1$ である場合には、仮想ノードを挿入して $N_{rmax} = N_1$ を満足させる。例えば、 $N_{rmax} = 3$ の画面領域を持った端末について考える。今図 3(a)のメニュー分類木が与えられている時、 $N_1 = N_{rmax} = 4$ にするために、一時的に仮想的な親ノード etc を同階層に作成し、その配下にあふれたノード d とノード e を再配置する(図 3(b))。次に、 L_1 の 1 つのメニューをプルダウンし

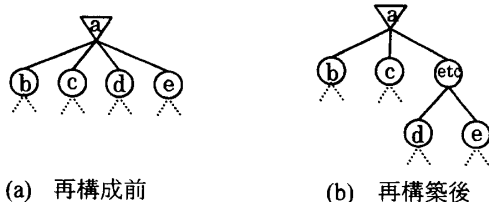
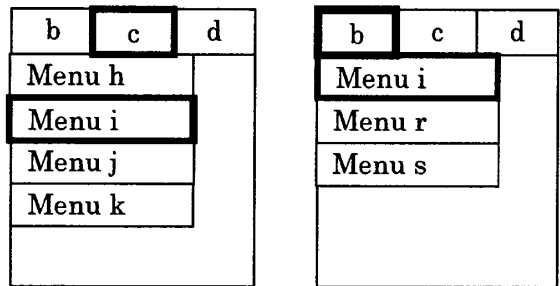


図3. メニュー分類木の再構築

た場合に、画面の縦の長さがプルダウン項目をすべて表示できない場合がある。すなわち、深さが 2 より深いノード例えば図 2 のノード i の子の数 N_i が N_{cmax} より大きい場合である。この場合にも仮想ノードの挿入を行ない、 $N_i = N_{cmax}$ を満足させる。最後に、プルダウンメニューをレベル i までプルダウンした場合に、各レベルをプルダウンメニューの右隣に表示すると N_{rmax} を超えてしまう場合がある。つまり図 4(a)のように迎ってきたメニュー階層をすべて表示できない場合、選択した親ノードメニューを残し、その他のノードを非表示にする(図 4(b))。



(a) ツールバー項目選択時 (b) 次階層表示

図4. プルダウンメニュー階層表示

メニューボタンエリアには、優先度を指定されたノードのうち優先度の高い順に N_{rmax} 個のノードを木の行きがけ順に表示する。ここで表示された分類ノードをユーザが選択したときには、その分類ノードの子孫ノードの中からさらに優先度が高い順に N_{rmax} 個を選択して表示する。図 2 のメニュー分類木に基づいてメニューボタンを表示する例を図 5 に示す。

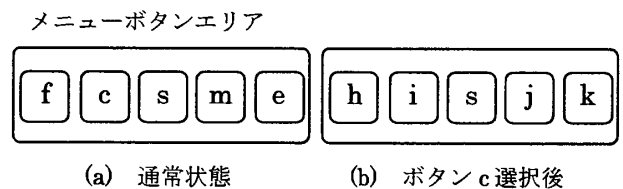


図5. ボタンメニュー表示例

5. おわりに

本報告では端末の HW プロファイルと端末に依存しない UI 構成情報に基づいて、UI を描画するためのフレームワークを提案した。今後ワークスペースを定義し、コンテキストメニューを組み合わせること、HWプロファイルとしてポインティングデバイスなどの情報を記述し、それに応じてメニュースペースを適応させることなどを考えている。またプロトタイプを実装して有効性を評価する予定である。

参考文献

[1] Karin Coninx et.al, "Dygames: Dynamically Generating Interfaces for Mobile Computing Devices and Embedded Systems", Human-Computer Interaction with Mobile Devices and Services, 256-270, 2003.
 [2] Mario Bisignano et.al, "Dynamic user interface adaptation for mobile computing devices", The 2005 Symposium on Applications and the Internet Workshops, 2005.