

複数の適応方法を考慮したAUIシステム

7M-4

長崎 等 東 基衛
早稲田大学理工学部

1 はじめに

本研究は個々のユーザに適したユーザインタフェースの構築を目的とする。今回は様々な適応要求に対して複数の適応方法を併用する適応型ユーザインタフェース（AUI）の実現方法について発表する。

2 AUIを実現するためのシステム

現在、我々は小規模なバイナリレベルでのソフトウェア部品を動的に組み合わせることで従来のアプリケーションと同様な機能を提供するコンポーネントウェア環境GEISHAを開発中である。この環境にユーザインタフェースの適応及びシステムの適応を実現するための仕組みを組み込んでいる。

2.1 基本的な考え方

AUIの実現に以下に述べるGEISHAの基本概念が適用可能である。

まず、第一にユーザインタフェースの独立である。これにより、ユーザインタフェースを変化させてユーザに適したユーザインタフェースを構築することが可能となる。

次に、アプリケーションのコンポーネント化である。アプリケーションを独立させることによって、つねに固有のユーザのためにカスタマイズされたユーザインタフェースを複数のアプリケーションで使用可能となる。

上で述べた基本概念に次の概念を追加することによって適応を実現する。

様々な適応に同時に対応可能な機構である。この機構があることにより、様々な適応をスムーズに組み合わせることができる。

また、このシステムでは適応のためのデータとして、ユーザのアプリケーション使用履歴データによるものを中心に考えている。

2.2 実現のための基本方針

この環境においてはタスクを記述することによってアプリケーションとして振る舞いを定義することができる。つまり、ここで考えるアプリケーション作成とはタスクの記述とタスクの実行に必要なコンポーネントの作成（または、選定）及びユーザインタフェースの定義である。

図1の部分は我々が開発中のGEISHAの適応を実現するためのメカニズム部分である。

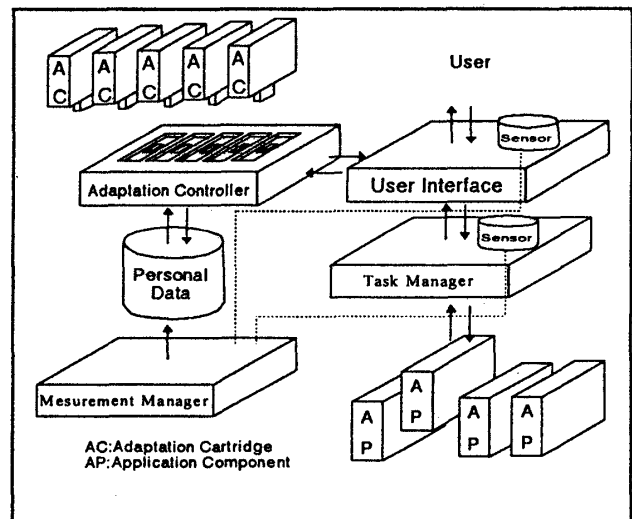


図1：適応のためのメカニズム

3 機能を実現するための各コンポーネント

User Interface

直接、ユーザから見えるユーザインタフェースである。これは、Adaptation Controllerの示唆により適応をおこなったユーザインタフェースであり、その状態は、個々のユーザによって異なるし、その使用時期によっても異なる。User Interface ManagerによってユーザインタフェースのInstanceが生成される。

Task Manager

ユーザが使用するタスクがTask Description Documentとして存在しそれをこれが解釈しシステムの制御を行う。

Adaptation Cartridge (AC)

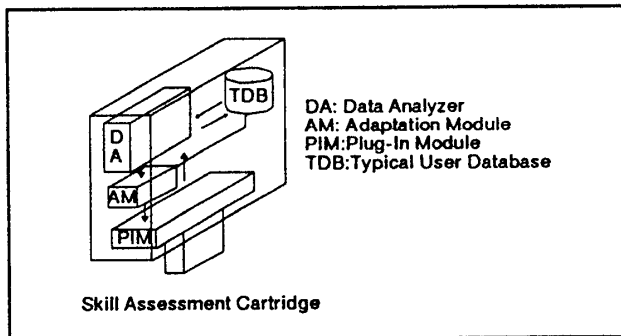


図2：Adaptation Cartridgeの例

個々の適応技術を実際に実行する部品である。このカートリッジをAdaptation Controllerにはめ込むことによって適応を実際に行う。カートリッジは適応の種類によって、複数存在する。

カートリッジの内部は、Adaptation Controllerとのインタフェースを保つPlug-in Moduleと適応機構からなる。

図2にあるのはカートリッジの内部の大まかな構造の例である。

Adaptation Controller

この機構によって複数の適応技術をコンフリクトなしにユーザインタフェースに反映させる。カートリッジにある適応技術によって、実際にユーザインタフェースを変更する部分である。コントローラを変更することによって、よりコンフリクトのないユーザインタフェースが実現可能である。またカートリッジ間の整合性の取り方に特徴があるユーザインタフェースが実現可能である。

Mesurement Manager

必要に応じてSensorをアクティブな状態にしたり、操作履歴などの書き出しタイミングを制御する。

Personal Data

個人の操作履歴が蓄積されている。他に今回は関係ないが内部の制御履歴なども含まれる。

Application Component (AP)

ComponentwareにおけるComponentであるが特にユーザが使用するアプリケーションの機能を持つComponentのことをさす。これがアプリケーションのもつ機能を提供し、単数もしくは複数のApplication ComponentとTask DocumentsとUser Interface Description Documentsから従来のアプリケーションに相当するものとなる。

Sensor

各コンポーネント、特にユーザインタフェースのInstanceやTask Managerなどに埋め込まれている。

4.4 適応のまでの手順

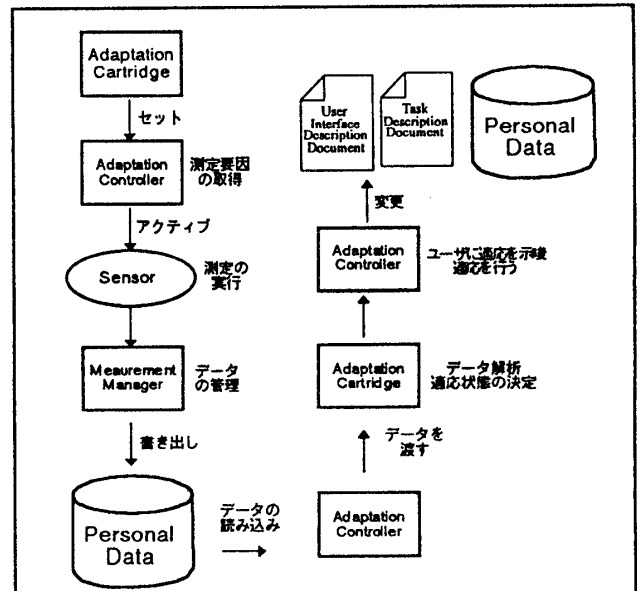


図3：適応までの手順

User Interface, Task Manager等にデータ収集用のSensorを埋め込み、ユーザの操作履歴を記録する。そのデータをMeasurement managerを経由し、Personal Dataとして保持し、Adaptation Controllerが各カートリッジの内容に沿って適応を行う。

5 開発中の環境下における適応メカニズムの特長

従来の適応型システムは単一の適応方法を前提としているものがほとんどだが、この環境では最初から適応方法の部分をカートリッジ化することによって多様な適応方法をサポートする事ができる。

タスク記述を変更することによって、システムの振る舞いを容易に変更できる。このため自己適応及び選択適応が容易である。

User Interface Description Documentを変更する事によってViewを容易に変更できる。同様に自己適応及び選択適応が容易である。

Application Componentの変更が容易である。このことよりシステムの自己適応及び選択適応ができる。

6 今後の課題

開発した環境を用いての適応実験や複数のAdaptation Cartridgeの開発などがあげられる。