ユーザに合わせたモダリティが選択可能なインタフェース・ツールキット

6U－1

川合 史朗，相田 仁，齊藤 忠夫
東京大学 工学部

1 はじめに

計算機－ユーザ間のコミュニケーションに、文字情報だけでなく画像情報や音声情報を用いることは普通に行なわれるようになってきた。それらの様々なモダリティを利用したアプリケーションは、多くの人にとっては親しみやすく使いやすい物である。しかし、ユーザの使用環境によっては、そのようなインタフェースが最適ではない場合もある。

例えば普段オフィスで使い慣れたアプリケーションを処理能力の低い携帯端末でも使いたいが、凝ったGUIよりは多少見かけを悪くしてもレスポンスの方を速くしてほしいという要求はあるであろうし、視覚障害をもつユーザにとっては、精細なグラフィカルユーザインタフェース(GUI)よりも音声化しやすいキャラクタベースのアプリケーションの方が使いやすいということもある[1]。

本稿では、これらの問題点を踏まえて、ユーザの要求に合わせてインタフェースの表現を切り替えるようにするためのツールキットfruit (Flexible and Rearrangeable User-Interface Toolkit) の構成について述べる。

2 ユーザの使用環境とインタフェース

計算機の使用形態が多様化するに従い、どのようなインタフェースが最適かということは一概には言えなくなってきている。インタフェースに影響を与える要素は、図1のように分類できる。

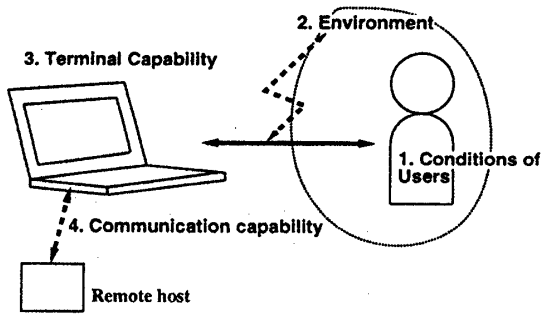

図1: インタフェースに影響を与えるファクタ

1. ユーザの身体的状況：一時的／永続的な身体障害など
2. (主として一時的な) 使用環境に関するもの：周囲の騒音、振動や、移動しながらの使用など
3. 端末のケーパビリティ：限られた画面やその他のデバイス、限られた計算能力
4. 通信路のケーパビリティ：データソースやアプリケーションそのものが遠隔地にある場合の通信路容量との兼ね合い

はじめから考えうる全ての環境に適応できるインタフェースを作ることは困難であるので、まずは次のような形態のインタフェースを実現することを考える。

- 上記1, 2, 3に関して：同一のアプリケーションで、状況に応じてGUI, キャラクタベースユーザインタフェース(CUI)*、音響的ユーザインタフェース(AUI)†を選択して使えること。
- 上記3, 4に関して：リモートでアプリケーションを起動している場合、端末とアプリケーション間の通信量を必要に応じて増減できること。たとえば、遅い回線だから画像情報は送らない、等。
- 上記2に関して：アプリケーションの使用中にも環境は変動するので、必要に応じて動的にインタフェースの形態を切り替えられること。

3 アーキテクチャ

インタフェースツールキットfruitは、ユーザの環境に合わせてダイナミックにインタフェースのモダリティを切り替えることを可能にするツールキットである。従来のインタフェースツールキットを用いたアプリケーションと、fruitを用いた場合との違いを図2に示す。

Fruitはcommunication stub, interaction shellおよびsession managerの3つの部分から構成される。

"An Interface Toolkit with Dynamic Selectable Modality for Individual User Requirements"
Shiro Kawai, Hitoshi Aida and Tadao Saito
Faculty of Engineering, The University of Tokyo

*CUIにも、キャラクタ端末のフルスクリーンで自由にカーソル移動を行うインタフェースと、UNIXのシェルのように行単位の入出力を行うものとがある。

†本稿では、AUIはauditory outputとkeyboard inputの組み合わせを指すものとする。将来は音声による入力も一般的になるであろうが、現時点での視覚障害者の計算機使用形態を考えると入力はkeyboardと考えるのが妥当であろう。ビンディスプレイによる出力は一種のCUIと考えられる。

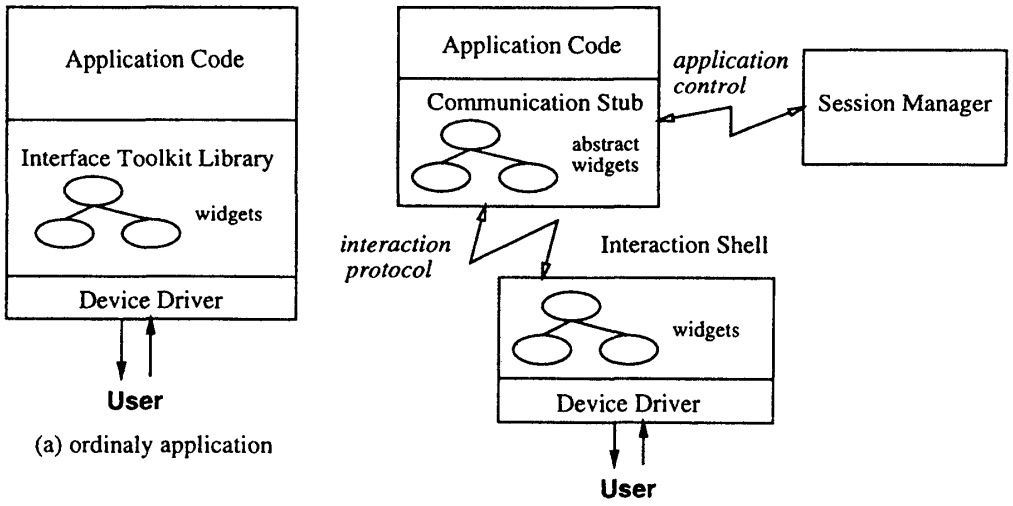

図 2: Fruit を用いたアプリケーション

Communication Stub

アプリケーションとリンクされるライブラリ。普通のGUI ツールキットと同じように、インタフェースの部品となる widget というオブジェクトの API が提供され、アプリケーションプログラマはそれを用いてインタフェースを記述する。

しかし、アプリケーション側では widget に関するデータを保持しておくだけで、それを具体的に表現するメソッドはアプリケーション側にはない (このため、 communication stub 側からみた widget を abstract widget と呼ぶ)。 Communication stub は interaction protocol を使って widget の作成や状態変化の情報を interaction shell とやりとりする。

アプリケーションは interaction shell と一時的に切り離すこともできる。次に別の interaction shell と接続したとき、インタフェースの表現を再構成するための情報を communication stub は保持している。

Interaction Shell

実際に widget を適切なかたちで表現し (rendering)、ユーザとのインタラクションを担当するのが interaction shell である。一般に、一人のユーザが計算機に向かっているとき、ひとつの interaction shell が立ち上がって、ユーザとのすべてのインタラクションを処理する。 Lexical, syntactic なフィードバックは interaction shell で行なわれる。

Widget をどのような形態で表現するかは interaction shell の実装に任される。

Session Manager

アプリケーションが interaction shell とは独立しているため、すでに立ち上がっているアプリケーションと interaction shell とを接続したり、アプリケーション自身を起動したりするために session manager というデーモンプロセスが用意される。

Abstract Widget

Communication stub の提供する widget は GUI ツールキットの widget と似たようなものだが、各々の部品がインタラクションの意味をあらわすという点で GUI における widget とはやや異なる。例えば、ファイル名を入力するためのインタフェースは、 GUI では普通 label, textentry といった widget とそれらを配置するための widget とを組み合わせて作られるが、 abstract widget ではそれは「短いテキストを入力する」というひとつのオブジェクトになる。そのため、 GUI の場合はラベルに使われる文字列が、 CUI や AUI ではプロンプトとして表現されるなど、インタラクションの意味に即した表現が可能になる。

4 おわりに

インタフェースを表現する部分とアプリケーションとを分離し、実行時に必要なインタフェースに切り替えて使えるようなシステムを作成した。今後、従来のインタフェースツールキットからの移植のしやすさや、 interaction protocol の性能の評価などを行なってゆく予定である。

参考文献

[1] G. C. Vanderheiden. Building Disability Access Directly into Next-Generation Information and Transaction Systems. In *Proceedings of the IISF/ACMJ International Symposium on Computers as Our Better Partners*, pages 2–6. World Scientific, Mar. 1994.