

GUI 部品 of 交換によるデザイン変更のための プラグインアダプタ

2J-2

佐藤 博之, 増田 英孝, 笠原 宏
東京電機大学 工学部

1 はじめに

いくつかの GUI 部品には同じ操作目的を持つものがある。どの部品を使用するかを決定するのは、アプリケーションプログラマである。しかし同じ操作目的であれば、どの部品を使いたいかはユーザ毎に異なる。ユーザから部品交換の要求があれば、プログラマがアプリケーションに変更を施し、ユーザへ還元するのが現状である。UI の独立性により複数の UI 戦略が利用可能となっているが、それは設計段階においてである。部品交換がユーザレベルで実行時に可能となれば、より柔軟なインタフェースを提供できる。ユーザレベルでデザインの変更が可能なのは GUI の 1 つの目標であり、それを様々な形で積極的に促進すべきである。

そこで本研究ではユーザレベルでのデザインの変更を容易にすることを目的として、GUI 部品ごとの接続インタフェースを整合させ、いくつかの基本部品で構成された GUI 部品群として利用できるアダプタを提案する。実例としてラジオボタンをボタン群と捉え、単一選択リストと同じ接続インタフェースを持った部品、ラジオボタンアダプタを作成した。また同様に、複数選択を行える複数選択リストに対しチェックボックスアダプタも作成した。

2 GUI 部品の変更

2.1 接続インタフェースの不整合

ライブラリユーザ（アプリケーションプログラマ）が、GUI 部品を使ってユーザインタフェースを作るとき、操作目的によって部品を選択する。そして類似した目的の部品は容易に交換可能であると便利である。しかし実際には、各部品の接続インタフェースが異なっているため変更は容易ではない。その接続インタフェースは、むしろ部品の内部構造が似ているものごとに類似性がある。表 1 に示したように、ライブラリユーザにとっての類似した部品とライブラリアンにとっての類似した部品の捉え方が異なっている。それはライブラリアンの作成した部

品で、内部構造の似たもの同士の実装が似てくるのは自然であり、その方がライブラリアンにとって負担が少ないためである。

2.2 単一選択リストとラジオボタン

例えば単一選択リストとラジオボタンという GUI 部品がある。図 1 に単一選択リストを、図 2 にラジオボタンを使用した、同一の単純なアプリケーションを示した。これらはどちらも単一選択をするための部品である。単一選択をしたい場合、どちらを使いたいかはユーザによって異なる。そこでこれらが容易に交換可能であるとすれば、ユーザが使いたい方を選択すれば良いことになる。

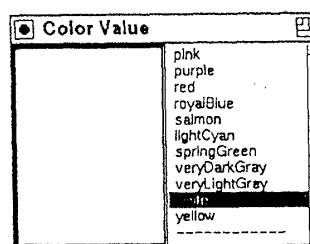


図 1: 単一選択リストを使用したアプリケーション

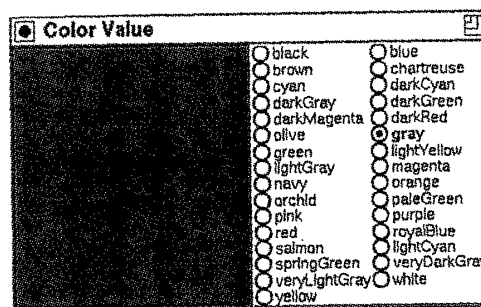


図 2: ラジオボタンを使用したアプリケーション

ユーザが使いたい部品を自由に交換可能とするためには、それらインタフェース部品とアプリケーション本体（モデル）との接続方法 [1] が、同等であれば問題はない。しかし、単一選択リストの接続方法とラジオボタンの接続方法が異なっているため困難が生じる。図 3 にリストの接続方法を、図 4 にボタンの従来の接続方法の概念図を示した。リストはそれ自身で 1 つの部品であるが、ラジ

表 1: ライブラリユーザから見た部品の実作目的とライブラリアンから見た部品の内部構造

		ライブラリユーザ	
		単一選択	複数選択
ライブラリアン	リスト	単一選択リスト	複数選択リスト
	ボタン群	ラジオボタン	チェックボックス

ボタンはボタンという構成要素が組み合わされて作られており、このため単体の構成要素であるリストとボタン群の交換には都合が悪い。ボタンという構成要素1つ1つを用いて作成されたボタン群という構造で捉え、その接続方法に整合性を持たせれば問題は解消される。

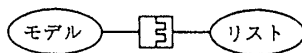


図 3: モデルとリストの接続図

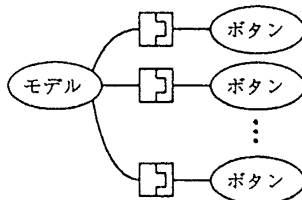


図 4: モデルとボタンの接続図

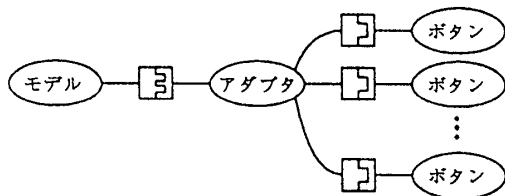


図 5: モデルとラジオボタンアダプタの接続図

そこで本研究ではその整合を取ったラジオボタンアダプタを作成した。その接続の概念図を図 5 に示した。接続方法は従来のボタンとは異なるが、実際には同じ部品で構成されているので使用法は変わらない。

同様のことが複数選択リストとチェックボックスの間にも指摘できる。こちらも同様にチェックボックスアダプタとして作成した。

2.3 アダプタの評価

リストにはリストの、ボタンにはボタンの利点がある。ボタンの場合、配置やラベルを工夫しユーザの認知性を高めることができる。

今回作成したものはボタンのこのような特徴を活かしきれていない。ボタンは、リストと同じような配列で上から下へ並べたものである。ユーザの認知性を高めることができる、ボタンとしての特徴を活かせるものがより望ましい。しかしリストが使用されているものの代りに、このアダプタを使用することは簡単にできることが、第一の特徴である。今回のようにアプリケーション本体(モデル)から見て、リストもボタン群も1つの部品と見ることができるとは、部品接続に一貫性を持たせた点で重要な特徴である。これによりアプリケーションプログラマは自由に部品を選択できるようになり、Metamer [2] を用いることでエンドユーザによる部品交換の可能性も生まれた。

3 まとめ

エンドユーザによるデザインの変更が可能であるという GUI の1つの目標を達成するために、ラジオボタンアダプタとチェックボックスアダプタを作成した。これにより、リストとボタン群をアプリケーション本体(モデル)と接続する方法に一貫性を持たせることができた。そしてアプリケーションプログラマは自由に部品交換ができるようになり、エンドユーザによる部品交換の可能性も生まれた。しかしそのアダプタは、配置やラベルを工夫してユーザの認知性を高めることができるという、ボタンの特徴を活かし切れていない。今後はそのような点を改善していく予定である。

参考文献

- [1] Objectworks\Smalltalk release4.1 User's Guide. ParcPlace Systems, Inc. (1992)
- [2] 増田, 笠原: Metamer のユーザ履歴管理, 情報処理学会第 48 回全国大会 2J-1 (1994).