

木構造モデルによる インタフェースとロジックの分離に関する研究

柳田拓人* 栗原正仁

(北大情報科学)†

1 はじめに

今日、交通機関のチケット予約、書籍の通信販売といったサービスが広く利用されるようになり、また外出先から操作可能な情報家電製品などが登場してきた。しかし現状のアプリケーション(サービス)には、それぞれ別々の操作方法を習得する必要があったり、様々な使用環境に対し GUI しか提供されていないという問題がある。

ところがアプリケーションごとにインタフェースを実装する従来の方式では、すべての操作方法を統一したり、アプリケーションの提供者が複数のインタフェースを用意することは困難である。従来研究として、インタフェース記述言語 UIML [1] や、GUI を文字情報に変換する研究 [2] があるが、問題解決には至っていない。

そこで本研究では、複数のアプリケーションでユーザが好みのインタフェースを統一して使え、また状況に応じて様々なインタフェースに取り替えられるようにすることを目的とした、新しいアプリケーション・システムを提案し、その実装例を示す。

2 提案システム

従来のアプリケーションにおけるインタフェース上のインタラクションをモデル化し、提示選択インタラクション・モデルと称する。そしてそれに基づいた記述を提示選択インタラクション・モデル記述と称する。また、アプリケーションをインタフェースとそれ以外の部分であるロジックとに分離、連帯させ、それぞれインタフェース・クライアントとロジック・サーバと呼ぶ (Fig. 1)。

提案システムでは、ユーザが用意するクライアントのレンダラが、サーバと共有したインタラクション記述よりユーザが操作するインタフェースを生成する。サーバ側ではインタフェースの代わりにインタラクション記述を提供する。双方は接続時、互いにインタラクション記述を操作する。レンダラは、ユーザによるインタフェースの状態変化と記述の変化を対応付け、またサーバはロ

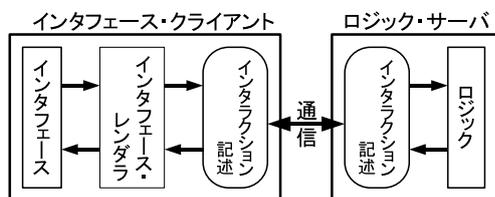


Fig. 1 クライアント・サーバ関係

Table 1 インタラクション記述言語の要素

コマンド名	解説
構造要素	Interaction (インタラクション), Session (意味上の大きなまとまり), Section (意味上のまとまり), Ordered (順序付けフラグ), Immediate (即時通知フラグ)
選択要素	BinaryChoice (二値選択), ExclusiveChoice (排他的選択), RangeChoice (範囲内の数値選択), FreeChoice (選択肢のない選択), Command (動作の実行)
提示要素	Name (名称), Hint (ヒント), Explanation (説明), Description (詳細情報), Message (ユーザへのメッセージ)

ジックの内部状態の変化と記述の変化を対応付ける。レンダラが異なれば、単一の記述から異なったインタフェースが生成されるため、本研究の目的が達成される。

3 インタラクション記述言語

インタラクション記述言語は提示選択インタラクション・モデルに基づいてインタラクションを記述するための言語であり、情報構造および情報の選択と提示を要素とした木構造で表現される。この木構造をインタラクション木という。

言語において、モデルより示されたインタラクション要素の種類、すなわち情報構造および情報の選択と提示はそれぞれ構造要素、選択要素、提示要素と呼ばれ、それらの持つ意味によってさらに細かい要素に分かれる。これらをまとめてインタラクション要素という。要素は 0 個以上の子要素を持つことが可能である。

要素の種類

インタラクションの要素は構造要素、選択要素、提示要素である (Table 1)。構造要素はインタラクションの構造を表現する。Interaction, Session, Section は意味上の包含関係を表し、レンダラはそのことがユーザによって理解されるように表現する。Immediate はサーバ側がユーザの操作を即座に通知して欲しいことを表す。

選択要素はユーザの選択行為を表現する。二値選択を表す BinaryChoice, 排他的選択を表す ExclusiveChoice, 一定範囲内の数値を選択する RangeChoice, 選択肢のない選択を表す FreeChoice, 動作の実行を表す Command

*takty@main.eng.hokudai.ac.jp

†札幌市北区北 14 条西 9 丁目北海道大学大学院情報科学研究科

の5種類がある。またこれらの要素の子要素となって選択肢など表現する要素が存在する。

提示要素は名称やヒントといった選択要素の説明用の情報を表現する。Name, Hint, Explanation, Descriptionはそれぞれ説明の詳しさの度合いが異なる。Message以外の各要素はユーザの希望などによって、レンダラがいずれかの要素をユーザに示す。Messageはサーバからユーザへのメッセージを表現する。

4 木構造同期プロトコル

提案システムではクライアントとサーバがそれぞれインタラクション木を持つため (Fig. 1), 別々に操作されるこれら2つの木を一致させる必要がある。そのための手法が木構造同期プロトコルである。クライアントとサーバによるインタラクション木の操作情報は、それぞれが持つ操作履歴に保存される。そしてこの情報は特定のタイミングで両者間で交換され、両者の木が等しくなる。接続中における木の操作はクライアントからサーバへのコマンドで発生する。

プロトコル運用の流れ

クライアントと接続している間、サーバ側が木を操作できるのは、クライアントの ALL メッセージと GET メッセージを受け付けたときのみであり、操作は受け付けたメッセージの戻り値としてクライアントに送信される。

クライアント側ではユーザがインタフェースを操作することによって木が操作され、この情報はサーバと同じく操作履歴に保存される。対象要素が Command の子要素でなく、かつ Immediate を持っていない選択要素の子要素の場合、操作情報は履歴に保存される。一方、対象要素が先の条件を満たしていない場合、その操作の情報も含めて操作履歴がすべてサーバへ送信される。

以下にクライアント側から見た通信の流れをステップごとに説明する。

- Step 1 HELO メッセージを送信し、サーバから接続許可を意味する OK メッセージを受信する。
- Step 2 ALL メッセージを送信し、サーバ側木のすべての要素の情報を受信する。そしてその情報を元に木を更新し、インタフェースを生成する。
- Step 3 インタフェースの操作が木の操作となり操作履歴に保存される。即座に送信しなければならない選択行為によるものである場合、SET メッセージで現在の操作履歴を送信する。
- Step 4 SET メッセージの後すぐに GET メッセージでサーバの操作履歴を取得し、受信情報を元に木を更新する。更新した後 Interaction の子要素が残っていれば Step 3 へ戻る。
- Step 5 QUIT メッセージを送り、通信の終了を要求する。サーバは終了を許可するなら OK メッセージを返信する。

5 実装例

提案システムの実装例を示す。処理系に Borland C++ Builder 3, 通信方法として TCP/IP を使用した。

GUI によるインタフェース・クライアントを実装した (Fig. 2)。BinaryChoice の表現をウィンドウのサイズに合わせて自動的に変えるように、提示の詳しさをユーザが自由に切り替えられるようにした。また、卓上電気スタンドのシミュレーションとなる実装例を作成した。これに接続したクライアントからのスイッチの ON・OFF や明るさの切り替えが可能である (Fig. 3)。

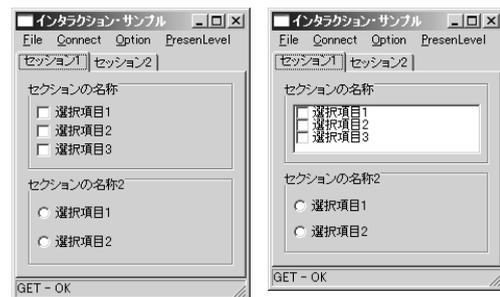


Fig. 2 GUI 版インタフェース・クライアント実装例の画面 (左が通常時, 右はウィンドウが小さくなったとき)

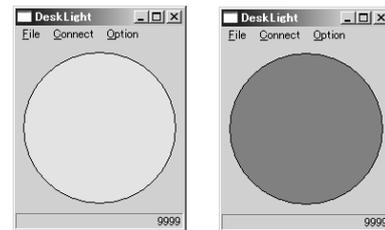


Fig. 3 ロジック・サーバ実装例の画面 (左が ON 時, 右が OFF 時)

6 おわりに

本研究では現状のアプリケーションの問題点である、アプリケーションごとにそれぞれ別々の操作方法を習得する必要があることや、ユーザの様々な使用環境に対し GUI が提供されていないことを解決するために、アプリケーションからインタフェースを分離する手法を提案し、実装可能であることを示した。

今後の課題として、本研究で作成したインタラクション記述言語の表現力の評価、提案システムを実際に運用した上での有効性の評価、音声入出力版インタフェース・クライアントの作成、評価が挙げられる。

参考文献

- [1] Harmonia. "Tutorial Booklet December 1997."
- [2] 岡田 英彦, 旭 敏之. "ユーザインタフェース変換に基づく PC 遠隔操作システムの開発," ヒューマンインタフェース学会論文誌 Vol.4 No.4, 2002.