7Q－2　スレッドを用いたXWindow Systemのコールバック連結手法

遠城　秀和
NTTデータ通信（株）　開発本部

1. はじめに

近年、高精彩なビットマップディスプレイを用いたより使い易いユーザインタフェースとして、グラフィッカルなユーザインタフェース(GUI)が種々の分野で活用されている。特にGUIを構築する業界標準的ツールとしてX Window Systemが普及し、効率良くユーザインタフェースを作成出来つつある。しかし、GUIに適した作成手法は十分でない。

本報告では、ユーザインタフェースを構築する上で、操作手順の限定有無両方の組み合わせが必要であり、X Window Systemで用いられているコールバックメカニズムが操作順序を限定した操作組み合わせを実現する際の問題を示す。さらに、スレッドとシリアライゼーションを用いた解決方法を提案する。

2. 操作順序の依存性

ユーザインタフェースは、コマンド指定操作、パラメタ設定操作、データ入力操作など複数の操作によって構成される。そしてユーザは、コマンド指定操作とそのコマンドが必要とするパラメタ設定操作を行うというように幾つかの操作を組み合わせてシステムを利用している。

操作の組み合わせ方には、以下の2種類がある。

(1)操作順序が限定された組み合わせ

特定のコマンド指定操作を行い、次にそのコマンド特有のパラメタ指定操作を行うというように操作順序が限定された状態で操作の機能が定義される。

(2)操作順序が限定されない組み合わせ

各コマンド共通のパラメタ設定操作とコマンド指定操作というように操作順序に関係なく各操作の機能が定義される。

操作順序が限定された組み合わせの場合、操作間の依存性が高くなる。前の操作ごとに次の操作が定義され、操作数が多く使いにくいユーザインタフェースなりやすい。操作順序が限定されない組み合わせの場合、操作間の独立性が高くなる。操作を組み合わせる自由度が大きくなり、数少ない操作で使い易いユーザインタフェースを構成しやすい。

しかし、特殊なコマンドに対する特異なパラメタ設定操作のように独立にすると逆に使いにくい場合もある。操作順序が限定されない組み合わせだけでユーザインタフェースを構成することは、使い易いインタフェースの実現を困難にする。従って、GUIでも2種類の操作組み合わせを実現可能とする手法が必要である。

3. コールバックメカニズム

X Window Systemでは、画面上のマウスクリックによって生成されるイベントとそのイベントに対応する処理を関係付ける方法としてコールバックメカニズム[1]が用いられている。コールバックメカニズムは、イベント発生時に行う処理をコールバック関数として用意し、イベントにその関数を登録する方法である。イベントが発生すると登録されたコールバック関数が独立に起動実行される。（図1）

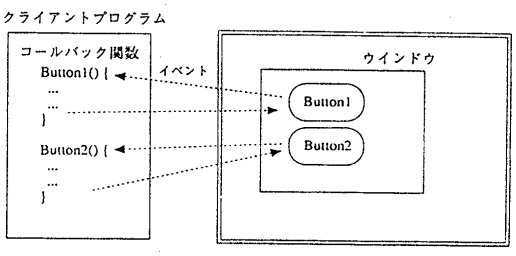

図1　コールバックメカニズム

イベント発生からコールバック関数の起動実行までが順序独立な振り分け処理（図2）として標準提供される。このため、コールバック関数を作成するだけで操作順序に依存しない組み合わせを容易に実現可能としている。

しかし、コールバック関数間での並行処理機能がなく、コールバック関数の中で入力待ちなどの処理中断をすると、振り分け処理に戻れなく次イベントを処理できない。また、イベント待ちをコールバック関数中で使用するとイベント待ちがコールバック関数中と振り分け処理中の2箇所になり、競合が発生し問題となる。操作順序を限定した組み合わせの実現には、コールバック関数中の処理中断とイベント待ちを可能とする必要がある。

Threaded Callback Serialization for the X Window System Programming
Hidekazu ENJO
NTT DATA COMMUNICATIONS SYSTEMS CORPORATION

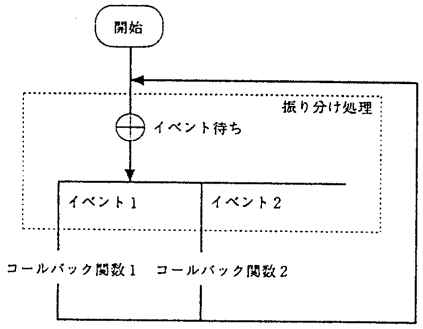

図2　振り分け処理

4. スレッドを用いたシリアライゼーション

スレッドを用いると並行処理機能をプロセス中に実現出来る。コールバック関数起動をコールバック関数を実行するスレッド生成、起動に置き換えると、コールバック関数中で処理中断可能となる。（図3）

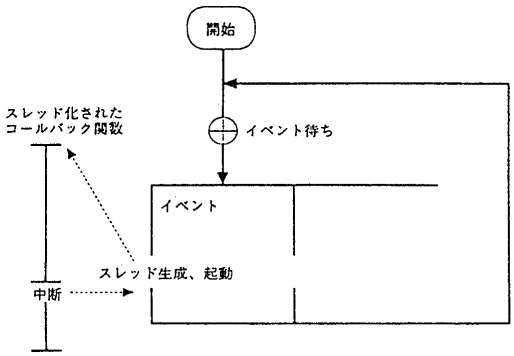

図3　スレッドによる並行処理

コールバック関数中でイベント待ちを実現する方法として、複数コールバック関数を順次実行し、連結したコールバック関数間をイベント待ちに見せる方法（シリアライゼーション）が考えられる。具体的には以下のように実現する。（図4）

(1) まず最初のイベントに実際のコールバック関数を実行するスレッド生成、起動処理を登録する。

(2) そのスレッド中でイベント待ちを実行する場合、現在のイベントの登録を一時的に削除し、自スレッドの再開処理を待つイベントに一時的に登録する。その後、自スレッドを中断する。

(3) 待っているイベントが生成されると一時的に登録していた再開処理を削除し、スレッド再開処理を実行する。その結果、中断していたスレッドが再開する。

(4) このように次々に関数の登録と起動とを行い、コールバック関数の処理終了後、最初のイベントの登録を復旧する。

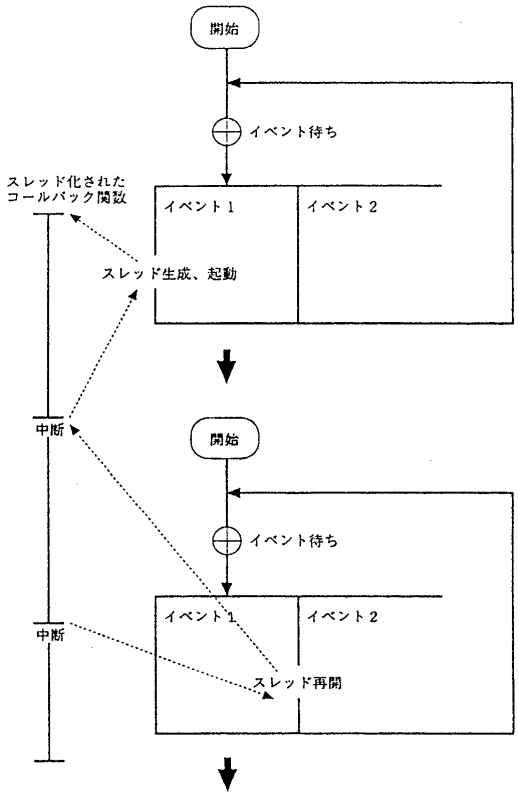

図4　シリアライゼーション

5. まとめ

本報告では、ユーザインタフェースを構築する上で、操作手順の限定有無両方の組み合わせが必要であり、X Window Systemで用いられているコールバックメカニズムが操作順序を限定した組み合わせを実現する際の問題を示した。

さらに、スレッドと複数のコールバック関数を順次実行し、あたかもコールバック関数を連結しているように見せるシリアライゼーションを用いた解決方法を提案した。

参考文献

[1] Nye, A. and O'Reilly, T.: "X Toolkit Intrinsics Programming Manual", pp25-28, O'Reilly & Associates,Inc(1990)