8E−2

CONCURRENT WELL−PPP におけるサービス管理

平井 郁雄　丹羽 直人　守屋 洋　村尾 洋　榎本 肇

芝浦工業大学

1．はじめに

並行実行型画像システム記述言語（Concurrent WELL-PPP）の開発にあたってのサービス管理とは、この言語を使うユーザが、煩雑な手続きを踏むことなく、システム内部でのデータの授受を行えるようにし、更にそれらのデータを保存管理してやることと、Request-Respondの関係を持った関数実行の手続きを整備してやることである。オブジェクト指向の立場で考えると、このサービス管理は、データを取り扱うための1つのオブジェクトと考えることが可能であり、我々は、このオブジェクトをデータ・マネージャと呼ぶ。共同開発を進めていくにあったては、こうすることにより、明確な分担作業を行うことが出来る[1]。

そのデータ・マネージャの中で、データを一括して扱う都合上、データのフォーマットをテンプレートの形で導入した。その結果、データの扱いがより明確になり、データ・マネージャとファンクション・マネージャの間のやり取りやデータ・マネージャとウィンド・マネージャ間のデータのやり取りが一元的なものとなり、それぞれのクラスへのデータの流れが簡略化された。

2．Concurrent WELL-PPPシステム内でのデータ・マネージャ（DM）の位置付け

Concurrent WELL-PPP は、プロトタイプから始まり、後での機能拡張を考慮に入れ、オブジェクト指向に沿った開発が進められている[2]。そのため、機能をそれぞれのクラスへ分解していく作業は、慎重に行われる必要があり、実際、そのように行った結果、図1のようにまとめられた。

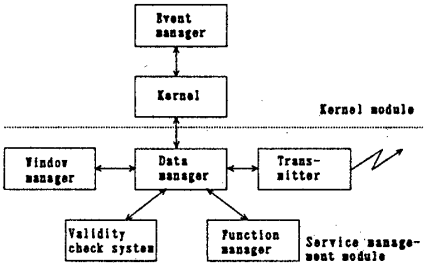

図1　Concurrent WELL-PPPの構成

DMは、カーネル（K）、トランスミータ（TR）、データ・ウィンド・マネージャ（DW）、ファンクション・マネージャ（FM）、バリディティー・チェック・システム（VCS）の4つのクラスとつながりがあり、TRを通して2台目のワークステーションとメッセージの授受を行うことで並行処理を可能とする。もちろん、テンプレート情報がWS2へ送られた後は、WS1とは切り離された形で処理を進めることが出来る。

3．データ・マネージャの役割

サービス管理の中核であるDMの具体的な役割について述べる。第一に、DMは、先ほどのシステム構成図からも明らかなように他のクラスとの窓口的役割を果たしている。カーネルより送られてきた関数実行要求は、テンプレートの定義準備、定義操作を行い、それにあわせて、関数の実行結果を表示するための要求をウィンド・マネージャへ、データの転送要求をトランスミータへ出すという形で処理される。

第二に、DMは、テンプレートの定義準備を行わなくてはならない。これは、つまり、どのような手順でテンプレートを準備するのかということである。これについては、後の節で詳しく述べるが、基本的な考え方としては、縦軸（Y軸）方向に、その高低差（値域）を取り、それから1本の線に対して用意するテンプレートの数を算出するというものである。このように定義することによって、どのような関数実行要求に対しても同じ処理で済ませることが出来、保守性を向上させることが出来る。

第三に、定義操作である。DMは、定義準備によって用意されたテンプレートに値を入れるために、カーネルから受けた関数を実行させて、それを行う。

4．Request-Respondで表されるインタラクション

テンプレートを必要な数だけ関数実行前に用意する定義準備（Defining Process）と用意されたテンプレートに具体的に値を代入する定義操作（Define Operation）は、DMにおいてカーネルから実行要求があった場合、1セットで機能し、カーネルとのインタラクションをRequest-Respond の関係で表すことが出来る[3][4]。具体例として、データ・ウィンドに点を打つ場合と主要点に色をのせる場合を図2、図3に示しておく。

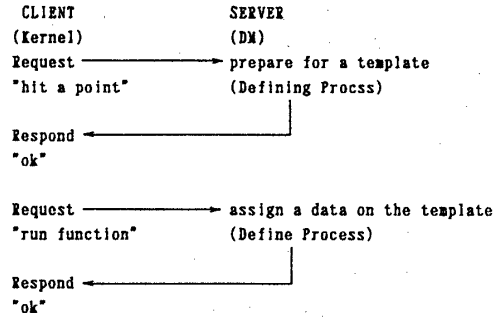

図2　DATA_WINDOWに点を打つ場合

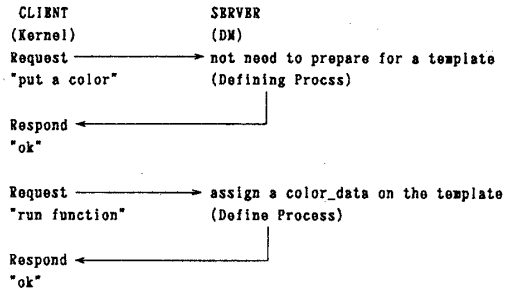

図3　主要点上に色をのせる場合

図2、図3より明らかなように、システム内部で、カーネルは、DMに対してクライアントとして作用し、DMはサーバーとして作用しており、1回のプロセスでDefining ProcessとDefine Operationが必ず1セットで働くこと

Service Manegement for Concurrent WELL-PPP
Ikuo Hirai, Naoto Niwa, Yo Moriya,
Yo Murao, Hajime Enomoto
Shibaura Institute of Technolozy

からクライアントとしてのカーネルとサーバーとしてのDMは最低２回のやり取りが行われ、カーネルは、どこまで作業が進んだかを常に把握した状態にいることが出来る。

５．テンプレートの定義準備と定義操作との整合性処理

まず、前提条件としてWELL-PPPで開発されるPicture Painting System は、画像を横方向（Ｘ軸方向）に取った水平走査線を基本的な単位として考え、この水平走査線１本分を１つのリージョン・セグメント（Region Segment ）とするという定義を持つ。また、更に、そのリージョン・セグメントを複数の領域に分割したものをサブリージョン・セグメント（Subregion segment）とする。これは、Color-Section での色補間作業に関係がある。つまり、色をラインとラインの間で補間するときには、サブリージョン・セグメントごとに補間作業を行ってやるからである。

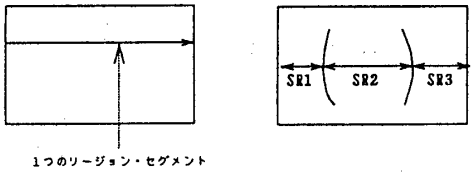

図４　リージョン・セグメントの定義

これらの定義により、色の補間作業がＹ軸単位にＸ軸方向になされてゆくことがわかる。つまり、テンプレート情報は、１つのラインに対して各Ｙ軸上に基本的には、１個づつあればよい。これは、主要点のＹ座標の値域によって簡単にテンプレートの必要数を算出できることを表している。

５．１．単調性曲線の場合

単調性のある曲線は、必ず次の３つのケースに分類される。①単調増加　②単調減少　③傾きなし　①②のケースでは、単純にＹ軸方向にその値域をとれば、準備すべきテンプレートの数が分かる。下の図５では、値域が２５である。実際の点の数は主要点も含めて、２５＋１＝２６個である。

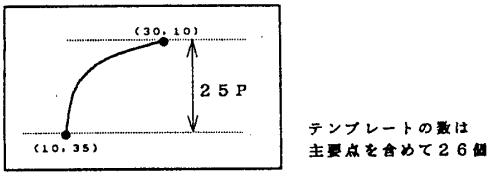

図５　単調性曲線の場合の算出法

また、例外的に図６のようにＹ軸方向に変化がない場合、つまり、③（傾き０）の時は、主要点の間に新しくテンプレート情報を置く必要はない。

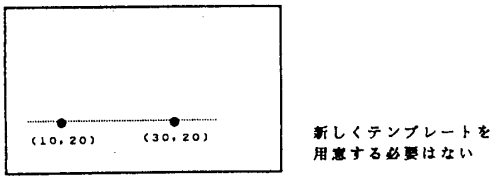

図６　単調性曲線の例外

また、直線は、単調な増減線の特殊なものとして捉えることが出来、同じ手順でテンプレートを準備することができる。

５．２．非単調性曲線の場合

非単調な曲線は、単調な場合と異なり、単調な増減として表すことは出来ない。しかし、今、Defining Processの手順として、Ｙ軸の値域を考える事で必要なテンプレートの数の算出を行うので、非単調なものに関しては、特別な扱いを行わなければならない。その特別な扱いとは極値点を分割点として、１本の曲線を複数の単調増加、減少な曲線に分ける線分割の考え方である。例として、図７のような曲線を示す。左は、９点を打って、それをスプライン関数で接続したものであるが、複数の単調増減する曲線に分解すると右のように４つのラインに分解される。

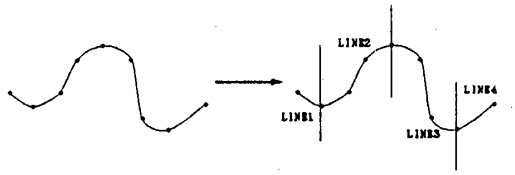

図７　非単調性曲線の分解例

しかし、これには、１つの問題点がある。ユーザが与える主要点をスプライン関数で結んだ曲線の極値点がその主要点と重なるとは限らないということである。例えば、図８のような場合である。

図８　非単調性曲線の問題点

この場合、極値点に新しく主要点としての登録を行いラインを分解していく。つまり、スプライン関数起動時に、左から点の情報を得ているとするならば、この関数の中に極値点を判別するためのチェック機構を織り込んでやり、極値点である事がわかったならば、そこでテンプレート情報を操作して、１本目の分割線を完成させ、更に、Defining Processでもう１度テンプレートの準備をした後に続きの接続を行ってゆくのである。

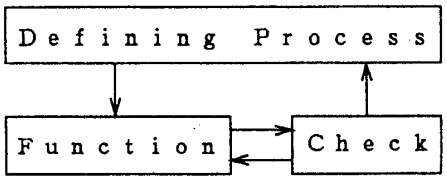

図９　極値点発見のための手続き

こうする事で、全ての曲線に対して線分割が可能となり、Defining Processの簡易性、保守性が維持される。

６．まとめ

Concurrent WELL-PPP を使う事によってユーザは、テンプレートを使ったPicture Painting System を複雑なウィンド間のやり取りや図形の情報管理を省いた形で実現する事が可能となる。また、内部的にも、サービス管理を独立させることによってカーネルの構造を簡単で明確なものに出来る。これは、図形編集に限らず他分野への応用も可能であり、これから、更に標準となってゆくウィンド・システムのエンド・ユーザ対策として大きな意義がある。

文　献

[1] 鴨志田、丹羽、榎本：”オブジェクトネットワークによる画像描画システム記述言語”、情報処理学会第４４回全国大会、１９９２．３
[2] トッパン　J．ランボー他：オブジェクト指向方法論OMT、１９９２
[3] 守屋、丹羽、村尾、榎本：”協調型画像描画システムの並行図式”、情報処理学会第４６回全国大会、１９９３．３
[4] 丹羽、守屋、村尾、榎本：”並行実行型画像システム記述言語（Concurrent WELL-PPP）の実現”、情報処理学会第４６回全国大会、１９９３．３