

# Extensible WELL における対話的動作制御システム

1 Y-8

宮城利文 青木 稔 村尾 洋 榎本 肇

芝浦工業大学

## 1 はじめに

研究・開発が進められている拡張機能言語 Extensible WELL(Window-based ELaboration Language)を用いたシステムでは、多種類のサービスに対応し、User とシステム間及びサービスプロセス間の協調的作業を円滑、かつ効果的に進めるため、インターフェースの役割をするシステムが必要である。すなわち、処理システムの動作状態を的確に監視、表示、制御するシステムが必要である。本研究ではこのシステムを対話的動作制御システムと呼び、そのシステム構成を明確にする。

## 2 動作制御システムのサービスの対象

Extensible WELLを用いたシステムで協調処理を対話的に行なうことによって効果を上げるよう意図しているが、そのため必要な動作制御システムのサービスの対象は大きく分けて2つ考えられる。User がサービスを要求する際(図1参照)と複数のスペシフィック・ロール・サーバで協調処理を行わせる際(図2参照)である。それぞれの動作制御は2つの制約処理;User との対話によって作業内容を制約する形態制約 (Modal Constraint) と作業状態の完了のチェックなどを行なう時相制約 (Temporal Constraint) を対話的に行なうことによって実現できる。以下に、動作制御の主なポイントを項目別に示す。

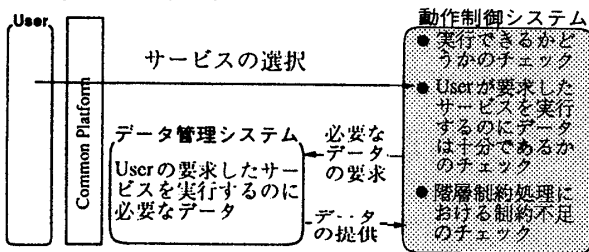


図1: 動作制御システムの制御の流れ (1)

### (1) User によるサービス要求時の動作制御

User がサービスを要求する際に、的確に作業を進めるために、操作を正しい方向に導くことが必要となる。例えば、動画像生成過程で、未だ静止画像を描いていない状態で絵を動かす要求を誤って出しても、動かす絵自体がないのでその要求をどこに出せばいいかわからないことになりかねない。そこで、User の要求が確実に実行できるサービスに要求意図を限定する必要がある。この処理システムでは、User が依頼したサービスをもとに総称的データ管理システム<sup>[1]</sup>やプロセス構築層<sup>[2]</sup>によって作業を行うプロセスの準備ができる。動作制御システムは現在のサービスの作業進行状態を把握するように構成できるため、可能なプロセスだけに実行許可を出す。これは、動作制御システムの形態的制約処理を行なう部分である。なお、サービスの作業進行状態はフラグをたてることにより把握できる。

Dialogic activity control system based on Extensible WELL  
Toshifumi Miyagi, Minoru Aoki, Yo Murao, Hajime Enomoto  
Shibaura Institute of Technology

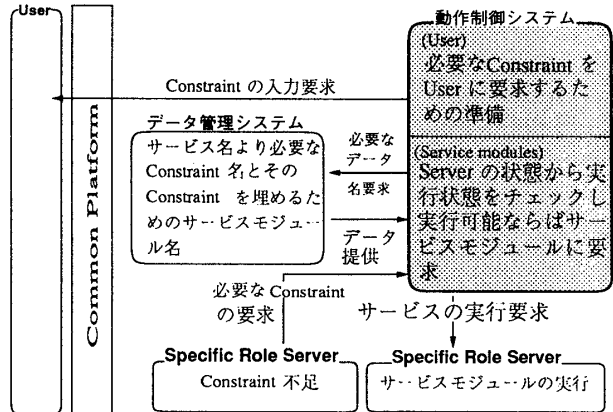


図2: 動作制御システムの制御の流れ (2)

### (2) サービス準備のチェックを行なう動作制御

User がサービスを要求する際に必要な制約として、User が指定したサービスを実行する準備が処理システムにできているかのチェックを行うことも必要である。Extensible WELLのシステムでは各分野に対応する処理プロセスにおいて、状態を名詞オブジェクト(ノード)、状態遷移の関数を動詞オブジェクト(プランチ)とする階層構造が存在するネットワーク<sup>[3]</sup>と呼ぶ。各サービスを実行しようとする際に、このオブジェクトネットワークの起動が完了しておらず作業ができない場合、プランチの処理が終了しておらず次のノードへの作業ができない場合などサービス実行不可能な場合が考えられ、動作制御システムがそれらをチェックし制御することになる。これらの制御も準備完了ならフラグをたてることによりチェックを実現できる。これは動作制御システムの制約処理の1つで時相的な制約処理を行なう部分である。

### (3) サービス実行時のデータが不十分である時の動作制御

User が要求したサービスを実行させるための Constraint が十分であるかをチェックする必要がある。例えば、動画像の例で2つの物体を衝突させる作業を行うサービスを考えると、物体をそれぞれ抽出させた後、それぞれの大きさ、動きデータ等の制約条件が必要となりそれらの制約条件がないと衝突させる処理を行うことができない。衝突させようとする際のイベント駆動で Constraint が不足していることが分かると、User もしくはサービスモジュール(それぞれの分野に対応する処理を行なう関数の集合)に要求することになる。もちろん実行するのに必要なデータが全てそろっていると動作制御システムは呼び出されることがない。一般に制約条件のチェックの方法は、動詞オブジェクトからデータ管理システムにそのサービスを実行するのに必要な制約条件を問い合わせ、それをもとにチェックを行うことである。

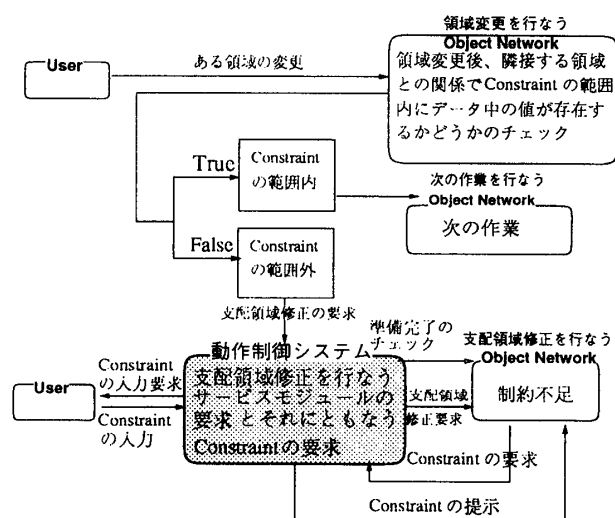


図3: 動作制御システムによる制御の例

#### (4) 階層制約における動作制御

Extensible WELLによる画像描画システムのオブジェクトネットワークに表現されているサービスプロセスの階層に基づき、上位にあるオブジェクトと下位にあるオブジェクトの関係から、下位のオブジェクトの新規操作によって上位のオブジェクトが修正される階層制約処理<sup>[4]</sup>を行なう必要がある。上位オブジェクト修正の際にデータ (Constraint) 不足になる場合が生じるが、動作制御システムはデータ不足を認識し、Userもしくはサービスモジュールと対話することによって必要なデータを要求することになる。

#### 3 動作制御システムの流れの例

2節で挙げた動作制御を用いて実際に制御する例を挙げる(図3)。この例は、顔画像など画像の変形に伴い生じる画像の不連続を修正するオブジェクトネットワーク<sup>[5]</sup>で作業を行なう場合である。作業に必要なConstraintを要求する作業において、Constraint作成の当事者はUserの例である。動作制御システムはUserが実行できるサービスを現在の状態から限定する(1)の動作制御を行なった後、顔画像の表情変化をさせるサービスを選択した場合で考える。このオブジェクトネットワークは顔画像中のいくつかの領域を変化させ、隣接領域との関係を見て不連続にならないように設定したConstraintの範囲内であるかどうかを判断することになる。ここで、移動量が少ないなど変更がConstraintの範囲内であると、次の作業を行うか新たなオブジェクトネットワークの作業を行うことになる。しかし、Constraintの範囲外であることが分ると画像の不連続などの問題が生じ、画像の凹凸による濃淡を分離している分割線によって生成される領域である支配領域の修正が必要になる。Constraintの範囲外であると、オブジェクトネットワークは動作制御システムに制約処理つまり修正が必要であることを伝える。動作制御システムは領域変更を行なうオブジェクトネットワーク中の動詞オブジェクト(プランチ)から判断して、支配領域を修正するオブジェクトネットワークへ作業要求を出すことになる。この際、動作制御システムは時相的制約処理として支配領域を修正するオブジェクトネットワークの準備ができていないかのチェックをおこなう(2)の動作制御を行なうことになる。準備が完了していることがわかると支配領域修正を行うオブジェクトネットワークが起動されるが、支配領域の指定などの制約条件が必要となるので動作制御システムに必要なConstraintを要求することになる。動作制御システムは形態的制約処理

として、このConstraintをディスプレイ上のMessage Windowを介してUserと対話することにより要求し、実行するのに必要なConstraintを供給する(3)の動作制御をおこなうことになる。

#### 4 サーバ間のコミュニケーション

Userとサービスモジュール間のインターフェースとしての役割において動作制御システムが扱うべきデータを以下に挙げる。

- Userが要求したサービス名
- Userが要求したサービスの実行に必要なデータ(Constraint)
- 作業を行っているオブジェクトネットワーク名
- 現在のノード名
- 当事者(User, サービスモジュール)
- Constraintを作成するためのサービスモジュール名
- サーバが実行可能状態であるかを判定するためのフラグ
- Userが要求したサービスの実行に必要なデータは十分であることをチェックするためのフラグ

これらのデータは、動作制御システムが汎用的であるために必要最低限のデータである。それぞれのデータが動作制御システムでどのように使われているかを以下に示す。まずUserが要求したサービス名は、Userの要求つまりUserが行おうとしているサービスを動作制御システムが知るためのデータである。Userが要求したサービス名を認識することによって、そのサービスを実行するために必要なデータをデータ管理システムに要求する。このデータはUserが要求したサービスを実行するのに必要なデータとして十分であることをチェックするためのフラグと共にサービス実行可能かのチェックを行うことになる。2節でも述べたが、チェックの結果データが不十分であることがわかった場合は形態的制約処理として、Userもしくはサービスモジュールによりデータを供給または、作成してもらう。現在作業を行っているオブジェクトネットワーク名と現在のノード名は、サーバが実行できる状態であるかどうかを判定するためのフラグと共にサーバ上に他の処理を行っていたり、処理が完了していないかをどうかをチェックする時相的制約処理を行なうために使われる。当事者のデータは、データを作成する作業者のことで、データ不足の際当事者にデータを要求してもらうことになる。Constraintを埋めるためのサービスモジュール名は、当事者がサービスモジュールでデータ不足の場合、データ作成作業を行うことの可能なサービスモジュールに要求するためのデータである。

#### 5 まとめ

動作制御システムの基本構成を明らかにした。Extensible WELLの特性に対応し、汎用性のために必要なデータの種類、主な制御は2つの制約処理によって実現できることなどを導いた。これらにより、動作制御システムがUserとサービスモジュールとの間で対話を行い、双方の作業を効果的に行なうよう導くインターフェースの役割をするシステムであることが具体的に示された。

#### 文献

- [1] 田村, 青木, 村尾, 榎本 “Extensible WELLのオブジェクトによる総称的データ管理システム” 情報処理学会第52回全国大会, 1996.3
- [2] 堀, 青木, 田村, 村尾, 榎本 “Extensible WELLにおけるプロセス構築” 情報処理学会第52回全国大会, 1996.3
- [3] 青木, 村尾, 榎本 “拡張機能言語 Extensible WELLにおける協調処理の制約的实现” 情報処理学会第52回全国大会, 1996.3
- [4] 榎本, 村尾 “モデル駆動とプロセス間結合” 情報処理学会第52回全国大会, 1996.3
- [5] 菅間, 宮田, 村尾, 榎本 “連続性制約駆動に基づく画像修正” 情報処理学会第52回全国大会, 1996.3