Extensible WELLにおける協調処理

2L－3

青木稔　平井郁雄　宮城利文　堀淳志　村尾洋　榎本肇

芝浦工業大学

1 はじめに

拡張機能言語Extensible WELL(Window-based E-Labolation Language)[2]は，分野記述言語であり，ある特定の分野を指定すると分野記述言語になりうる言語のことである。またWindow指向言語でありClientとのInterfaceはCommon platformと称するWindowを介する。またオブジェクト指向言語であり，Nodeと呼ばれる名詞Objectとそれらを結ぶBranchと呼ばれる動詞Objectで継承，合成を実現する。このシステムはClient server modelであり，Common platformを介したClientとServerの対話を実現する。その際に個々のModuleは，それぞれ役割をもっており互いに依存しあいながら目的を達成する役割依存型システム[4]である。また新たに協調型並列実行システムとして実現され独立性を持つProcessは並列実行される。制約処理システムとしてProcess間には制約条件が付加される場合があり，その制約内容に基づいた処理が実行される。本論文では協調的並列実行とサービス間制約の管理の方法論について述べる。

2 Client Serverの機能

これまでExtensible WELLでは，Client(人)とServer(機械)との対話型Modelを構築してきたが，新たにその間に機械としてのClientとServer(機械)との対話機能を追加した拡張型Modelを構築する。この人にかわってServerとやりとりをする機械のClientのことをAgent[1]と呼ぶ。Agentは担当分野のExpertである。人は抽象的Process名を総称的Object network上で選択し，Agentが得意担当分野を計画する。Agentが人の要求意図を読みとり，サービスの割り当て計画を行う。個々のサービスはそれぞれ専門家に任せて，Agentは専門家から得た結果をまとめ，内容をチェックし人にわかりやすいように変更した上で人に評価をゆだねる。すなわち,Client,Agent role server,Specific role serverによってInteractionが行われる。

ここでClientにはInteraction・Process中の仕事の内容によりUserとExpertに分かれる。
Userは，ある目的意図によってシステムを使用して，サービスの結果を得る役目を果たす。

一方Expertは，システムを分野記述型システムとして計画設計し，そのシステムを実現して，Userに使用可能な状態にする役目を果たす。またServerもその役割においてAgent role serverとSpecific role serverに分けられる。基本的な構成は統一的なコンセプトに基づくが，サービスの対象が異なる。

Cooperative process in the Extensible WELL
Minoru AOKI Ikuo HIRAI Toshihumi MIYAGI
Atsushi HORI Yo MURAO Hajime ENOMOTO
Shibaura Institute of Technology

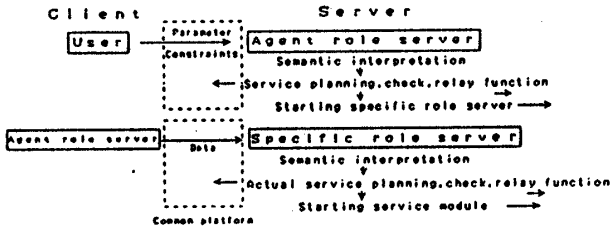


図１．Extensible WELLにおけるServer機能

Agent role serverはUserからの入力（パラメータ、制約条件）を解釈し、サービスの計画や結果の検査、及び中継機能を果たし、Specific role serverの起動を行う役割がある。一方Specific role serverはAgent role serverからのCommon platform上での入力を解釈し、具体的サービスの計画、結果の検査、中継機能を果たし、Service moduleの起動を行う役割がある。

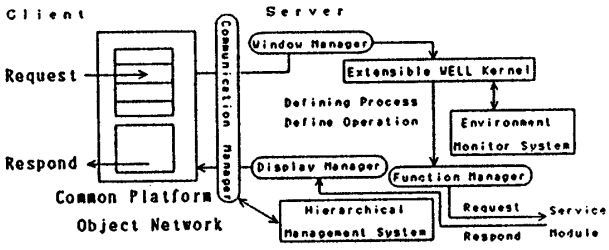


図２．Client Server間のinteraction

Clientの意図はRequestとして，Common platform上のGeneric object networkで入力され，Communication managerによって可視化対象である名詞Object名をインデックスに変換する。このCommunication managerの変換時に，必要があれば階層管理システムを利用して意図の細分化計画を行う。
Agent role serverではCommunication managerからの入力はWindow managerを経てKernelにつき，環境モニタシステムを利用し状態把握をした後，Clientの意図に対応する統合サービス用のテンプレートを準備(定義準備)する。その後KernelがFunction managerに実行要求を出し，Function managerがSpecific role serverにサービスのRequestを行う。Specific role serverでは，Agent role serverのFunction managerからのRequestを受けて，Communication managerがそのRequestの意味を階層管理システムを用いて分析し，Window managerを経てKernelにつく。ここで各サービスに対応するテンプレートを準備(定義準備)し，Function managerにKernelが実行要求を出し，Service ModuleにFunction managerがサービスをRequestする。Service Moduleがサービス結果をRespondするとテンプレートに代入(

定義操作)し，Display manager, Communication manager を経てRespondとしてAgent role server に返る。再びAgent role serverではSpecific role serverからのRespondを受けてAgent role serverのテンプレートに代入(定義操作)される。これがDisplay manager, Communication managerを通してRespondとして，データウインド上に可視化対象として表示されClientの評価を受けることになる。

3 協調処理

一般にUserの意図を満足させるには，複数のサービスを統合する必要がある。協調的並列実行システムを実現するためには，サービスプロセス間のタイミングを計る必要性がある。すなわちサービスプロセス間には時相制約[4]がある。そこでExtensible WELL では，Agent role serverがサービス計画をする際に，束構造を用いて並列実行可能プロセスに対しては，それぞれSpecific role serverに実行要求（Request）を出す。束構造における縦の関係は継承性を表し，横の関係は並列性を表す。各サービスの処理はSpecific role serverが担当し，Agent role serverがそれらを統合する。サービスに対応した束構造はExpertが知っており，Userはサービスプロセス間の時相制約について知っている必要はない。また，サービスの結果を継承する際には，形態制約[4]がある。これは並列的に処理された別々のObjectに対して，それらの関連性を示した制約のことである。これらの制約についての管理の方法論にも前述の定義準備，定義操作が当てはまり、同様のConstraint templateが用いられる。

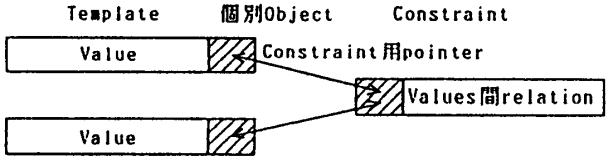


図３．データ値間制約

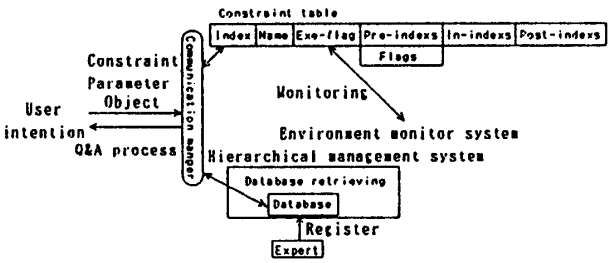
まずそのデータ値に対する制約を指すPointerを代入する領域をデータ値に付加したテンプレートを用語修飾節に基づいて個別Objectとして準備（定義準備）し，対応する制約Pointerの代入，実行処理（定義操作）を行い、参照されたConstraintの内容により妥当性チェックが起動されることで制約処理[3]が実現される。

4 環境モニタシステム

協調処理のためのサービスプロセス間のタイミングを計るには，各サービスプロセスの動作環境を監視する必要がある。この役割を果たすものが環境モニタシステムである。つまり各プロセスの実行状態（実行可能，実行中，実行不可）を把握するためのシステムである。このシステムは，動作環境を表す共有変数とそれを操作するプロシージャから構成される。各Serverにはひとつの環境モニタシステムが存在し，Serverの起動と同時にそのサービスに対応する環境モニタシステムが生成される。環境モニタシステムは，Object networkのNodeの把握，複数のObject networkの作業の完了の監視の２つの役割がある。Object networkの作業の終了は，Agentから階層管理システムの情報である終了すべきNode名を受け取っているので，そのNodeになると作業の完了をAgentに提示し，AgentはObject networkの作業が終了したことを確認することができる。

5 階層管理システム

ClientであるUserは，その意図をObject network上の名詞Objectによって入力する。Extensible WELLではこの名詞Objectの語彙構造を持っている。すなわち名詞Objectを意味論的に分析することが可能となっている。この語彙構造は担当Agent Expertが知っていて，基本術語についてのデータベースを持っている。階層管理システムはこのデータベースを管理し，名詞Object内の単語の階層性を管理している。また名詞に対応するObject networkを把握している。Userからの意図は，Communication managerを経て階層管理システムにつき，Expertによって登録された基本術語をもとにデータベース検索を行い，Single inheritance Multiple inheritanceといった情報を把握し、より詳細なサービス計画に変換されインデックス化され、テンプレートに代入される。



図．４　意図の詳細化

このテンプレートに基づき、Agent role server は様々なサービスを統合してUserの意図詳細化及びサービスの計画を行う。

6 まとめ

このようにExtensible WELLでは,Userの意図となる感性対象世界から形式的対象世界へ写像をCommunication managerが行い,そのUserの意図を満足させるために,Agent role serverが合目的構造モデルを構築する。またそれぞれ明確に立場の異なるClient,Agent role server,Specific role serverのinteractionの様子を示し，それぞれの役割に対応したinteractionの形態が示された。全体として並列的サービスの計画と統合のためにAgent機能を導入し，並列実行のための環境モニタシステムやUserの意図の詳細化のための階層管理システムにより協調処理が可能となる。

文献

[1] 榎本，村尾 " インタラクションの形態分析" 情報処理学会第５０回全国大会 2L-1,1995.3
[2] 富本，平井，青木，村尾，榎本，" 拡張機能言語Extensible WELLの体系化"，情報処理学会第４８回全国大会 4G-8,1994.3
[3] 平井，青木，村尾，榎本" Extensible WELLにおける制約処理"，情報処理学会第５０回全国大会，2L-2,1995.3
[4] 榎本，" 要求意図の対話的詳細化プロセス"，情報処理学会第４８回全国大会 4G-7,1994.3