

論理型言語向きサービスベースシステムの構成

6H-6

深沢 友雄・荻野 正・田中 英彦・元岡 達

(東大 工学部)

1. はじめに

サービスベースシステムは、網中に分散して存在する種々のサービス(データやプログラム)をユーザに提供する時に、各計算機の独立性を保ちつつ、かつ、ユーザには、分散性を意識させないユーザインタフェースを提供することを目的としている。サービスベースシステムでの計算機間の論理的な通信は、全て、サービスの要求と応答で行なわれる。現在まで、サービスの要求と応答を行なう言語として、関数型言語に基づく言語を用いるシステムの構成を研究してきた[1]が、今回、論理型言語に基づくシステムの構成について、検討と作成を試みたので、ここに報告する。

2. SBSにおける言語と論理型言語

SBS上で用いる主要な言語は、

- i) サービス定義言語
- ii) サービス要求言語

である、i)は、各計算機の提供するサービスを定義するための言語であり、ii)は、計算機-計算機間、及び、人間-計算機間で、サービス要求を伝えるための言語である。i), ii)をハイレベルな言語に設定することにより、ユーザインタフェースに優れたSBSを構成することができる。i)として、論理型言語に基づく言語を用い、サービスに関する定義を知識化すれば、ユーザは、簡単な要求で望むサービスを得ることができるであろう。i)とii)は、全く独立な言語でもよいが、使い易さ、処理効率、処理系の作成のし易さ等から、i), ii)とも論理型言語に基づくシステムの検討を行なった。

3. サービスの定義部と処理系

SBSにおける各計算機の主な構成要素は、サービス定義言語により、サービスの分散情報などを定義してある「サービスの定義部」と、外部(人間、あるいは、他計算機)からのサービス要求を処理する「サービスの処理系」である。本システムでは、サービス定義言語と、サービス要求言語として論理型言語を用いるので、SBS構成のためには、論理型言語に基づくサービスの定義方法[2]と、論理型言語に基づくサービス要求言語を解釈・実行する処理

系が必要となる。

4. サービスの分散と論理型言語

論理型言語向きSBSでは、サービスは、述語で定義される。SBSでは、サービスが分散して定義されているので、論理型言語における述語が分散して存在することになる。ユーザは、分散して存在する述語を自由に組合わせたサービス要求を行なう。この時の計算機間の通信の概念図を図1に示す。関数型言語では、サービス要求と応答が、関数呼び出しと、return-valueに自然に対応したが、論理型言語では、サービスの処理の結果の転送方法が問題となる。即ち、サービス要求は、ゴールの転送と自然に対応がつかうが、要求をした側は、処理の結果として、変数のユニファイされた情報と、真偽値の2種類の情報が必要となり、これらの転送が必要となる。(図1のS*, Q*等)SBSにおけるこれらの動作、即ち、サービス要求の解釈・実行と、計算機間の通信は、処理系が行なう。

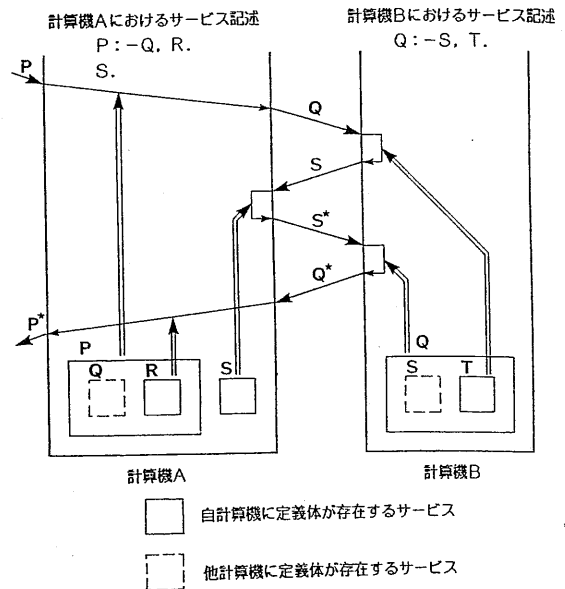


図1 論理型言語に基づく処理系間の通信

5. 処理系間の通信

5.1 SBSにおける計算機間通信

SBSのレベルでの計算機間通信は、

- i) サービス要求言語に基づく通信
- ii) テキストをそのまま送受信する通信

の2レベルに分かれる。ii)は、下位の通信機能とのインタフェース部であり、各計算機の通信機能に依存して作成される。i)は、サービス要求言語にのみ依存する部分であり、通信の最上位層となる。本節では、本システムに特徴的な部分である i) について述べる。

5.2 論理型言語に基づく計算機間通信

本システムの作成は、C-Prolog をもとにしたので、C-Prolog に基づく記法を用いる。Prolog による通信では、変数のユニファイに関する情報をどう伝えるかが問題となる。C-Prolog では、変数に直接アクセスできないので、変数を転送する時は、全ての変数をファンクタの要素として扱う方法を用いた。更に、解を返す場合、サービス要求(ゴール)の全ての解が求まってから解を返す方法と、1つの解が求まったらすぐに解を返す方法が考えられる。前者の場合、処理がとまらない場合が生ずることがあり、後者の場合は、通信回数が増加することになる。通信の簡便さの点から、今回は、後者の場合について検討した。

5.3 処理系の動作と定義部とのリンク

処理系が処理を進めている途中で、他計算機のサービスが必要となった場合、図2の形式のサービス要求・応答が行なわれる。他計算機で実行されるサービスも、全て、自計算機のサービス定義部に記述されている必要がある[2]。本システムでは、サービス定義時に、図3の様な定義を行なう。処理系で、externalという述語を用意することにより、サービス定義言語とのリンクを行なう。述語externalは、第1引数を図2-a)の<goal>として、サービス要求を行ない、図2-b)の形式の応答を順次、要求したゴールの変数とユニファイする。相手からの応答が、新たなサービス要求の場合は、サービスを処理して、結果を返す。計算機間をわたるサービス要求のネスティングは、任意のレベル可能である。

6. 考察・検討

6.1 変数に関する情報の伝えかた

図2 サービス要求・応答の形式サービス要求

- a) サービス要求...setof (varlist (<<goal>中の変数の並び>), <goal>, <解用変数>),
 ・ <goal>...他計算機で処理するゴール
 b) サービスに対する応答... [varlist (<solutions-1 >), varlist (<solutions-2 >), ...],
 ・ <solutions-i >...要求された<goal>の処理の結果ユニファイされた変数の並び

図3 サービスの定義部とのリンク

- <s-name> (<arg-list>) :- external (<es-name > (<arg-list>), <hostid>),
 ・ <s-name>...他計算機で定義されているサービスの呼び出し名 ・ <arg-list>...引数
 ・ <es-name >...他計算機上での当サービスの呼び出し名 ・ <hostid>...他計算機の名前

論理型言語に基づく言語での通信は、変数に関する情報の伝えかたがポイントとなる。本方式は、変数に関する情報をファンクタにして伝える方式を考えた。Prolog では、ゴール自身もファンクタなので、図2におけるファンクタvarlist のかわりに、ゴールそのものを変数情報を伝えるためのファンクタとして、転送してもよい。更に、図2の要求形式自体も、互いに全部の解を転送することがわかっているのだから、setof の形式で転送する必要はなく、単にゴールだけを転送すればよい。これは、要求と結果の転送をゴール単位で行なうことになり、Prolog の実行をゴールの書き換えとみると、書き換えたゴールを互いに通信しあって処理を進めるモデルと似ている。

6.2 通信の単位

解を返す単位として、全ての解集合という単位を考えたが、これにより、

- ・通信回数が減る。
- ・とまらない場合が生ずる。
- ・単独の計算機で逐次的に処理を行なう時と異なる解がでてくる可能性がある。

等の問題点がある。従って、解が1つである毎に解を返す様な通信方式もサポートする必要がある。

7. おわりに

本稿では、論理型言語向きSBS構成の為の主要部分となる、論理型言語に基づく計算機間通信の方法について述べた。今後、処理の並列化、OSなどのローカルな環境とのリンク、及び、この処理系の上でのSBSの応用システムの開発を行なっていく予定である。

< 参考文献 >

- [1]. 深沢、田中、元岡、「サービスベースシステムの概念と基本構成」、EC82-44。
 [2]. 荻野、深沢、田中、深沢、「サービスベースシステムにおける論理型言語向きサービス記述」、本大会、6H-7。