

サービスベースシステムにおけるサービス記述

1J-1

深沢 友雄 田中 英彦 元岡 達

(東京大学 工学部)

1. はじめに

計算機網上に存在する種々のサービス(ロード・モジュール、データ)をユーザに提供する時に、
 ・ユーザに対して、サービスの存在場所を意識させない。
 ・各計算機で独立にサービスの拡張ができる。
 事を目的としたシステムが、サービスベースシステム(SBS)である。

本研究では、SBSを構成する為の次の3レベルのビューを考えてきた。^[1]

- ①内部ビュー
- ②概念ビュー
- ③外部ビュー

本稿では、SBSの論理構成、各ノードでの各ビューの記述内容を述べ、実験システム上での記述方法を紹介する。

2. SBSの論理構成

SBSは、各計算機をノードとすると、図1の様な、有向グラフでモデル化できる。ここで矢印の向きは、サービス要求をする側から、サービス要求を受ける側に向く。各矢印について、矢印の先のノードをバックエンド計算機、元のノードをフロントエンド計算機と呼ぶ。各ノードについても、矢印が出ていく先のノードをバックエンド計算機、入ってくる側のノードをフロントエンド計算機と呼ぶ。更に、或るノードについて、そのノードから出ている或る矢印からの到達可能集合を、そのノードのバックエンドサービスベースシステム(BSBS)と呼ぶ。
 SBSでは、各ノードで、そのノードの全てのB

SBSと自分自身からなる計算機系を1つのシステムとして見せるビュー(外部ビュー)を、各フロントエンド計算機毎、或いは各ユーザ毎に定義する。従ってSBSでは、どの矢印から見ても、分散性を意識させないビューを提供する。この為に、各ノードでは、そのノードと、そのノードの全てのBSBSを統一的に管理する為のスキーマ(概念ビュー)を持つ。概念ビューは、各BSBSが提供するサービスの記述(BSBSの外部ビュー)と自ノードが提供するサービスの記述(内部ビュー)を用いて定義される。

3. 各ビューの記述

本研究では、図2の様な構成からなる実験システムを作成中である。このシステムでは、S-式による Lisp 的なサービス要求/記述言語を提供している。本章では、その言語に基いて、各ビューを構成する為の各サービスの定義方法を示す。

3.1 内部ビュー

内部ビューは、各ノード毎に独立に、それぞれのOS(VOS3, UNIX)上で定義する。サービスの実体は、各計算機のOSの制御下にあるデータやロード・モジュール、コマンド等である。即ち、各OSが提供するビューが内部ビューである。

3.2 概念ビュー

自ノードと各BSBS上にある全サービスを一意に識別できるサービス名を定義し、それらの、内部ビュー、もしくは各BSBSの外部ビューへの写像を定義する。各サービスの定義をする為に、次の2つの定義関数を用意した。

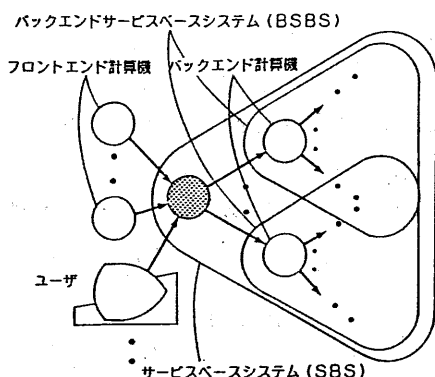


図 1 サービスベースシステムの論理構成

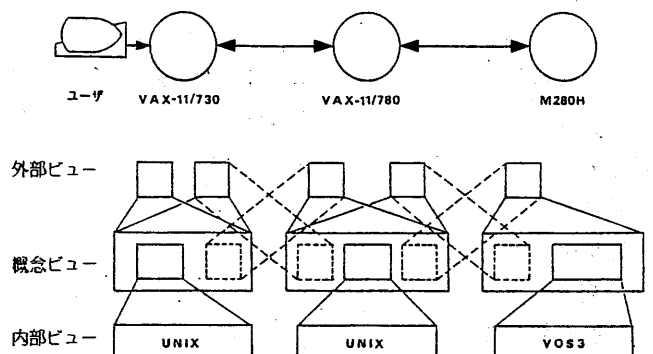


図 2 システム構成例

①自計算機のサービスの定義方法

(define-service

(<サービス名> <サービスの型>)<引数リスト> <定義体>)

<サービス名>…サービスの識別名

<サービスの型>…サービスの返す値の型

<引数リスト>…((<仮引数名> <型>) …)

<定義体>…内部ビューの呼び出し方

②BSBS上にあるサービスの定義方法

(define-backend

<サービス名1> <サービス名2><引数の渡し方> <場所>)

<サービス名1>…概念ビュー上での識別名

<サービス名2>…BSBSの外部ビュー上でのサービスの識別名

<引数の渡し方>…引数を評価してからサービスを起動するかどうか等

<場所>…BSBSの識別名

3.3 外部ビュー

概念ビュー上で定義されたサービスの部分集合を組合わせて作ったサービスの定義によって構成される。外部ビューは、各フロントエンド計算機或いは、そのノードからSBSを使うユーザ毎に定義する。

・定義方法

(defun (<サービス名> <サービスの型>)

(<引き数リスト>)<定義体>)

<定義体>…概念ビューへの写像をLisp風に書く。

4. ビューの内部構造

前章で述べた概念ビュー上のサービス記述を記憶する内部構造として、各サービス毎にLispシステムのPlistを用いている。Plistは、

((属性名 値) …)

という形式をしているので、サービス名をキーとすれば、データベースの関係モデルで表現できる。SBSでは、データのモデルとして、リレーションを取り扱っている^[2]ので、サービス記述自体をデータとして、SBS上のサービスで取り扱う事ができる。

実験システムでは、Plist上の属性として、

- ・場所
- ・名前※
- ・返す値の型
- ・引数
- ・引数の渡し型
- ・定義体

等がある。(※自ノードに実体がある時は内部ビュ

ー上の名前、BSBS上のサービスの場合、BSBSの外部ビュー上の名前)

外部ビューの定義は、Lispシステムにおける関数の定義機能を用いている。定義体の二次記憶系の管理は、各OSのファイルシステムを用いているが、将来は、概念ビュー同様リレーションとして表現し、SBS内のデータとして管理する。

内部ビューの定義は、各OSに依存しているが、この定義も、SBSに組み込む事により、概念ビューの記述をOSの構造から独立にすることができる。この時OS毎に異なるパラメータの指定の仕方 (optionalなもの、default 値、キーワード・パラメータによる記述等) をどう記述するかが問題となる。

5. 検 討

SBSでは、概念ビューの定義によりシステムが自動的にサービスの存在場所を見つけるので、計算機をSBSとして接続する時には、まず概念ビューを作成しなくてはならない。この時、他計算機上の各サービスについて3章の方法で定義するのは繁雑である。そこで以下の方法を検討中である。各BSBSに、一意に識別できる名前を定義する。(この名前は、定義しているノード内でのみ一意であればよい)次に各BSBSの外部ビューから、定義されているサービスの名前と、引数の渡し方等、概念ビューの定義に必要な属性値を持ってくる。(これは、外部ビューに対して、関係演算をほどこせばよい。)自ノードの概念ビュー上のサービス名として、BSBSの識別名とBSBS上のサービス名を結合した名前をつける。こうすれば、本ノード内での一意な識別名をつける事ができ、各BSBSが提供しているサービスに関する記述を自動的に生成することができる。BSBSに一意な名前をつけるにあたって、バックエンド計算機に関する様々な属性(ポート名、アクセス関数等)も、リレーションとして管理する機構を検討している。

6. おわりに

本稿では、SBSを構成する基本となる、サービスの記述について、主に概念ビューの構成を中心に述べた。現在、各ビューをリレーションとして管理するシステムの作成中である。今後は、障害対策のために、エラーに関するビューの設定等を検討していく予定である。

< 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」、信学技報、EC82-44.
- [2]. 矢部、深沢、田中、元岡、「サービスベースシステムにおけるデータ管理」、本大会、1J-2.