

3U-7

サービスベースシステムにおける
分散資源及び資源の仕様記述

何 千山* 橋高大造** 荻野 正* 田中英彦*

(*東京大学工学部 **三菱電機)

1. はじめに

計算機網上に存在する資源をユーザに提供する時に、
・ユーザは、サービスの分散性に煩わされることなく、網中のサービスを使用することができる。
・計算機網上の各計算機で独立にサービスの拡張が行える。
こと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。

本稿では、SBSの資源、及び資源の仕様記述について検討する。

2. SBSの分散資源

計算機の提供するサービスは、記憶機能と演算機能である
とってよい。すなわち模式的に、「作用と、作用の対象となるデータを与えることによって提供される機能」とらえることができる。

サービス = データ + 作用

ここで、サービス、データ、作用というものは全部資源として考えられる。作用とは、データをどう処理するかを記述したものである。通常我々が使用している計算機では、作用はコマンドに、データはファイルに対応している。サービスはデータと作用を組み合わせるものであり、サービスの実行とは、データに作用を施すことである。資源の記述はデータの記述、作用の記述、サービスの記述に分けることができる。

3. 資源の記述方法

SBSでは、ユーザは資源の利用方法さえ知っていれば提供されている全部の資源を利用することができることになるが、そのためにユーザの代わりにシステムが各資源についての情報を持っていることが必要である。ユーザがサービスを要求するときに、SBSがこの記述された情報に基づいて資源を組み合わせるユーザに提供する。SBSが資源をユーザに提供するとき、どのような情報を持っていないかについて考える。まず、資源が網の中に分散して存在するので、資源の存在する物理的な場所を知らなければならない。次に資源をアクセスする前に、資源の名前、資源のフォーマット、アクセス形式などを知る必要がある。これらの情報を資源の属性としてSBSに記述しておく。

属性間の関係から考えてみると：

(1). 属性の値として複数の値をとる場合がある。例えば、UNIXのコピーコマンドCPという作用を考えてみると、入力に関する情報という属性の属性値としてファイルができるし、ディレクトリもできる。

(2). ある属性が幾つかの従属属性からなっている。つまり、属性の値として再び属性とその値の構造体を持つネスト関係の場合がある。例えば前述の入力に関する情報がI/Oに関する情報の一つである。

以上の属性と属性値の関係をうまく表すために資源に関する情報をtreeの形式で持つことが考えられる。treeの形では、親ノードが複数の子ノードを持っているので、複数の属性値を持つような属性も表す可能であり、属性の値として再びこの形のsub-treeを持つてくることによりネストした情報を表すことも容易である。図1. に示すのはUNIXのCPコマンドという作用をtreeの形式で表す例である。

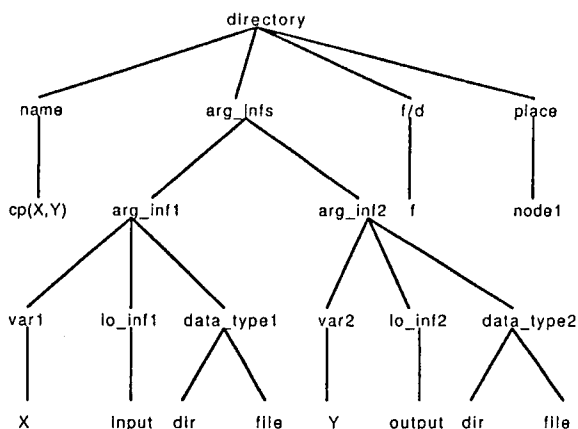


図1. 属性の関係を表す tree

これらの tree で表す属性を計算機に利用できる形式で記述するために既にリストの形式による記述 [1] を提案した。このリストの形式をSBSの本体記述形式という。ユーザがSBSに資源を要求するときにSBSはこのリスト中の属性情報を分析し資源を組み合わせるユーザに提供する。

ユーザが新しい資源を登録するときに、資源に関する各属性、つまり資源の仕様を知らなければならない。そして適当な言語を使って資源の仕様を記述する。もちろんユーザが本体の記述形式を言語として資源を記述することができるが、その形式は計算機が使える形式で、ユーザがわかりにくいし、実際にリストを作るのも複雑な仕事である。そこで、ユーザ

が資源を記述するために、より高級言語が必要となる。資源の仕様記述言語はそのために設計された言語である。

4. 仕様記述言語

仕様記述言語を設計するときに、まず言語に対する要求を考える。仕様記述言語を使用して資源を記述する人はSBSのシステム作成者ではなく、普通のユーザであるので、その読解容易性、作成容易性等が重要である。そして、全部の資源のメタ情報を記述できるようにしなければならない。資源のメタ情報というのは、資源の属性に関する情報の集りである。属性に関する情報としては、主に属性とそのとる値、属性間関係である。このようなメタ情報をうまく記述できるのは仕様記述言語に対する要求である。

4. 1. 仕様記述言語SDL

SDLは、前述の要求を満たすように設計されたものである。SDLは主に二つの部分に分けることができる。一つの部分はDICTIONARY情報の記述である。DICTIONARYは資源に関してユーザに提供する情報であり、資源についての機能や意味を記述したものである。計算機がこの情報を使わないので、自然言語で記述してもよい。ユーザがこのDICTIONARYの情報により自分のほしい資源を捜すことができる。

もう一つの部分はDIRECTORY情報記述である。DIRECTORY情報は計算機がその資源を使うために使用する情報である。DIRECTORY記述にはさらにDEFINITION部分、DESCRIPTION部分、SERVICE DESCRIPTION部分に分けられる。DEFINITIONには、資源の各属性名とその属性値を定義する。属性として資源の名前、存在場所、データのフォーマット、入出力の情報等があって、その中の必要なものだけ記述される。DESCRIPTIONには、各属性の関係を記述する。各資源に関する情報を記述、変更するときや、その情報を読み出して利用するときには、SBSが扱う全部の属性、属性間関係等を知っておく必要がある。DESCRIPTION部分には属性間関係をBNF記法で表し、左側を親属性、右側を子属性の並びとし、その意味は親属性がいくつの子属性からなっているという意味である。

資源に関して必要な属性の種類とこれらの属性間関係をシステムで予め記述しておく。属性の種類と属性間関係はそのシステムの固有のものであり、システムの知識の一部であると考えられる。ただし、今後、記述すべき属性が増えた場合や、現在考えている属性間関係よりもっと複雑な資源を記述する場合がある。この場合に新しい属性を追加したり、新しい属性間関係を定義したりする機能が資源の記述、登録者にとって必要な機能である。

以上の考え方によって、ユーザが実際に資源を記述する時には、必要な属性についてのみ記述を行えばよい。その中の属性と属性間関係を追加したり、変更したりしてもよい。

2.にも述べたように、サービスはデータと作用の組み合わせるものである。SERVICE DESCRIPTIONには、サービスの組み合わせ方が記述され、同じBNF記法を使い、左側を親とし右側を子とする。データと作用を記述するときにSERVICE DESCRIPTION部分が省略される。

図2.に示すのは、SDLでUNIXのCPコマンドという作用の仕様記述例である。

```
KEYWORD: cp(X,Y)
DICTIONARY: This function copy input X to output Y,X and Y can be
             directory or text file.
DIRECTORY:
DEFINITION {
    c_name::= cp(X,Y).
    e_name::= s_cp(X,Y).
    i_name::= cp(X,Y).
    place::= node1.
    f/d::= f.

    var1::= X.
    io_inf1::= input.
    data_type1::= dir | file.

    var2::= Y.
    io_inf2::= output.
    data_type2::= dir | file.
}
DESCRIPTION {
    directory::= c_name, e_name, i_name, f/d, place, arg_infs.
    arg_infs::= arg_inf1, arg_inf2.
    arg_inf1::= var1, io_inf1, data_type1.
    arg_inf2::= var2, io_inf2, data_type2.
}
END DIR
EOF
```

図2. 仕様記述例

4. 2. 仕様記述の作成ツール

- 仕様書専用エディタ：ユーザがSDLでSBSに新しい資源の記述、既存の資源記述の表示、変更等を行うときに、作成ツールが必要である。このツールの一つとしてSDLの仕様記述入力を表の形式等でガイドする仕様書専用エディタがある。専用エディタは主にデータ、作用、サービスなど三つの仕様書テンプレートからなり、自動メニュー生成、ヘルプ機能などを備える。

- SDLコンパイラ：SDLで記述された資源の仕様書をSBSの本体形式（リスト形式）に変更するコンパイラである。現在、専用エディタ、コンパイラをC言語でUNIXの上に作成している。

5. おわりに

本稿では、資源の仕様を記述する言語について述べた。現在、仕様記述のツールを作成している。ツールの完成後、多くの例を記述してみて、記述能力を評価していく予定である。

参考文献：

- [1]. 荻野 正, 田中英彦, 「論理型言語を用いたサービススペースシステム」IN86-130, 1987.