

記述の多様性を支援する記述名の名前解決法の提案

2 U-8

古宇田 フミ子

fumiko@mtl.t.u-tokyo.ac.jp

東京大学 工学部\*

1 はじめに

記述名は対象の属性や機能を説明する名前である。

このような名前が必要となる理由の一つは、異機種間の計算機結合システムにおける情報のやりとりのインターフェースとして、その資源対象の持つ属性を用いることができれば、一般的にも詳細にも、何らかの形で説明できることが多い[Go93]ことから、識別子の代替手段として便利なためである。

既に、記述名の表現法として、記述成分としての意味役割を表す簡単な構造を導入し構造化した記述名を提案している[F94]。図1と図2に示されるような表記法を持ち、属性、機能、使用法、識別子、環境の記述が可能である。記述表現の骨格は動作、関与者、周囲からなる vpc 構造と、一般名とその修飾語からなる em 構造として捉えられる。記述表現はこれらの組み合わせで表現される。

この特徴は、簡単な構造を持つことにより、記述表現の曖昧性が除かれ、記述の順序や省略に関して自由度を持ち、しかも、再帰的表現が可能なるものである。

このような記述名では、一つの対象に対する記述においても種々の記述が可能となるため、単に記述された対象を同定するだけでなく、異なる記述で同一対象を表しているかどうかの判定法や一つの記述表現に当てはまる同種の対象の決定法が問題となる。本論文では、この名前解決の問題について考察する。

2 記述名の表現例と多様性

例1 「情報分野の論文を探す。」という記述は、例えば、( search p( a paper of: information ) )... と表される。下線部の用語を変えることで「物理の本を探す。」等の表現が可能である。

例2 表現構造の違い: 知っている属性を示して、欲し

\*Conceptual view of a name resolution suitable for a descriptive-name notation with representative flexibility

Fumiko Kouda

Department of Information and Communication Engineering,

Faculty of Engineering,

University of Tokyo

3-1 Hongo 7-chome, Bunkyo-ku, Tokyo 113, Japan

```
SFe ::= clause | PA | id | us | Env      /* 記述側面 */
clause ::= ( { ST [PA] [CR] } )        /* 提供機能 */
PAe ::=
  p([A1][Partname] Ph [Attr:Md] [of:Ph] [(Partname clause)] )|
  P([A1][Agentive:] Ph [Attr:Md] [of:Ph] [(Partname clause)] )|
  p(Partname clause) | P([Agentive:] clause) /* 属性機能 */
CRe ::=
  c([Circname][Prep] Ph [Attr:Md] [of:Ph] [(Partname clause)] )|
  c( Circname[Prep] clause )          /* 記述要素 */
id( Env: Ph )                          /* 識別子 */
us( Env: Cmd )                          /* 使用法 */
Env ::= e( type noun )                  /* 環境的記述側面 */
Phe ::= noun | type | (type [op] noun )
      | ( type op number )            /* Phの基本要素 */
atre ::= adjective phrase              /* 修飾語, Mdの基本要素 */
Partname = |Affected:|Agentive:|Recipient:|Resultant:|
Circname = |Locative:|Temporal:|process:|Purpose:|
          Respect:|Contingency:|Degree:|
```

但し、[ ] の中は省略可能 { } の中では束縛 毎に語順は自由

図1: Basic Components of Descriptive Aspects

い属性を求めたい場合は、例えば、以下のような表現が可能である(図3参照)。表現1は vpc 構造の組合せによって、表現2は通常の文ならば、関係代名詞で表されるもので、再帰構造を用いた表現、表現3は欲しい情報を目的語として表している。

例3 同じ対象に対する異なる記述: 検索機能を持つ対象が結果を表示する能力も持ち合わせた場合、記述法として、知りたいものを表示するという表現も可能となる。即ち、

( search p( a paper of:information ) )... 又は、  
( show p( a paper of:information ) )... の一方だけで記述することも可能である。

3 名前解決

名前解決の定義は、例えば、ODP part2 で以下の様に述べられる。

```

P ::= Pe | Pa | Po | Pc
Pe ::= Basic Component
Pa ::= Pe Pe | Pa Pe | Pe Pa
Po ::= { Pe | Pe } | { Po | Pe } | { Pe | Po }
Pc ::= { ( Pa ) | Pe } | ( Po ) Pe | { Pe | ( Pa ) }
      | Pe ( Po ) | ( Pc ) ( Pc ) | { ( Pc ) | ( Pc ) }

```

図 2: Compound Components

表現 1

```

( search p( a paper of: information Attr:keyword )
  ( want p( author title of:( the paper ) ) ) ) ...

```

表現 2

```

( search p( a paper of:information Attr:keyword
  ( Affected: ( want p( title author ) ) ) ) ) ...

```

表現 3

```

( search p( a paper of:information Attr:keyword )
  c( Purpose:
    ( want p( author title of:( the paper ) ) ) ) ) ...

```

図 3: Different Descriptions

Name resolution: The process by which, given an initial name and an initial naming context, an association between a name and the entity designated by the initial name can be found.

記述名の場合は、2章で触れたように記述は対象の一面を捉えたものであり、また、同一対象の記述に異なる表現もあるので、上記の定義では不十分と考えられる。そこで、記述名の名前解決は以下のように定義する。

(1) 記述から対象を同定する。

(2) 与えられた記述から別の記述を求める。換言すると、ある対象に関する与えられた記述を利用して、その対象についての別の側面の記述を取り出すこと、である。

(1) については対象の同定はある環境下における識別子と対応つけることと見なせば、本質的には名前解決は(2)の定義となる。名前解決でどの側面の記述を取り出すかについては、記述の入力時に何らかの方法で明確にする必要がある。

#### 4 名前解決処理: 構成上の問題点

記述の多様性が生ずる原因は一つの対象についての可能な記述のうち、一部分だけが記述されているためである。名前解決の定義から記述の各種の側面を把握しているものが必要となる。そのため、一つの対象についての可能な記述を(概念的に)まとめたものを記述ベースとする。記述ベースでは、記述の各側面において、一つの対象につき、観察し得るすべての要素を網羅しているべ

きである。その表記法は、図1と図2に示されるような表記を用いると入力記述名との比較が容易になる。一つの対象に対するこの記述の集合は概念的には一行(LDBと呼ぶ)で表される。

記述ベースとLDBを導入したことで名前解決処理は、入力された記述名(IND)がLDBの一部の記述表現と対応がつく時、INDとLDBの一部の一致を調べ、INDがLDBの一部と一致した場合、同じLDBの別の部分を取り出す(図4参照)処理となる。

入力された記述名(IND)は記述に柔軟性があるので、名前解決処理で入力記述名がどのLDBの一部と対応が付けられるかが問題となる。入力記述の記述量によっては複数のLDBとの一致があり得る。このことからLDB同士の関係づけ、即ち、記述ベース全体の構成法が問題となる。

記述の多様性から同じ内容を表す表現が複数あり得るので、INDに対して、どのような記述を提示して何を求めたいかを正しく解釈する必要がある。

以上の考察から、名前解決処理では、少なくとも、以下の手順が必要になる。

(1) INDの解釈: どのような記述が提示され、どのような(側面の)記述を求めたいかを明確にする。記述構文から分かる場合(2章例1)と記述構文だけではすぐには分からず、用いられた用語の意味の解釈が必要な場合(2章例2)がある。

(2) INDで提示された既知記述とLDBとの一致(matching)を調べる。完全に一致した時はLDBの残りの部分から必要な記述を取り出す。INDの記述がLDBの記述の部分集合となる場合も同様とする。

#### 5 おわりに

多様な記述が可能な記述名に向けた名前解決法について考察した。この議論の精密化と具体化や評価は今後の課題である。特に、LDBの作り方、記述ベースの具体的設計、INDの解釈法は重要である。

##### 参考文献

- [Go93] A. Goscinski, Supporting User Autonomy and Object Sharing in Distributed Systems, ISADS'93, pp.100 - 106, 1993  
 [F94] F. Kouda, ICON-9, pp.493-498, 1994

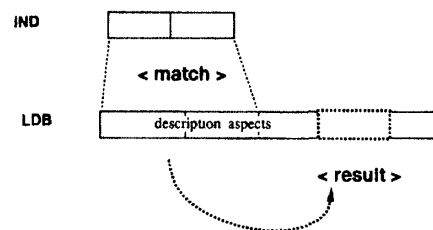


図 4: Concept of Descriptive Name Resolution