

記述名の利用者向き表現に対応した記述名管理のための 情報構造構成法に関する考察*

古宇田 フミ子[†]

東京大学 工学部 電子情報工学科[‡]

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

記述文の持つ意味構造に基づいた利用者向き記述表現法を仮定して、記述名の内部構造を構成した。利用者向き表現法では、記述の側面や記述の内容が多様に表現できる。内部構造では、この記述の多様性の支援と、多様性に基づいた名前管理向きの構成が必要になる。利用者向き記述表現法の特徴である and/or の等位節や節の再帰構造の可能性等を反映した記述側面の構造と、意味構造に基づいて分類した管理用情報ベースの構成法を考察した。

1 はじめに

記述名は属性等の記述により対象を識別するために用いる。記述名の使用目的の一つは、ある計算機システムで、利用者が使用法や識別子が分からない時に、他の属性や周囲状況の説明を用いて、それらの問合せができるようにすることにある。この場合、記述名に関する情報構造の流れを見ると、図1の形となる。これまでに、文としての表現を基にして、文の持つ曖昧性を除くために、構文と意味の関係を重視して、簡単な構造を持つ利用者向き記述名表現法を提案した。これは次章で触れる。本文では、3章でこの利用者向き表記法を仮定して記述名管理で必要な内部表現の構造について考察する。内部表現では、個々の利用者表記法に直接対応する構成と、各種の記述を集合として扱い、個々の記述を関連付けて統一的に管理できるような情報ベースの構成法に分けて考える。

ここで、記述名の考察範囲は、利用者向きで

*Internal structures corresponding to a notation of user-friendly descriptive names

[†]Fumiko Kouda

[‡]Department of Information and Communication Engineering, The Faculty of Engineering, University of Tokyo

「計算機」に関する資源、特に、各種 OS での資源の動的及び静的属性、識別子や使用法に限っている。

2 利用者向き記述名表記法の仮定

利用者向き記述表現を求めめるため、記述側面の見直しを行ない、記述可能な側面として、1) 対象の識別子、2) 対象自体の持つ機能（以後、属性機能と呼ぶ）、3) 他に与えることのできるサービス機能（以後、提供機能と呼ぶ）、4) 環境、5) 使用法、を取り出した。

これらの各々を記述側面と呼び、1)と5)を識別的記述側面、2)と3)を属性的記述側面、環境記述を環境的記述側面、と区分した。

ここで、環境は、対象の識別子や使用法が一意に識別可能な範囲を指す。

この各側面の記述のうち、属性的記述側面については記述の内容を文として捉えた。自然言語としての文には表現に曖昧性の問題がある。これを除くため、記述の内容を構文と意味の関係に基づき、簡単な構造を持つ表記法を提案した [1]。

利用者が頭の中で考えている記述 ⇔ 文としての表現 ⇒ 利用者向き記述表現法 ⇔ 対応内部構成
⇔ 管理用情報ベース

図 1: Information Flows

2.1 記述文に基づく属性的記述側面の記述の構造化

いくつかの例文を用いて、H. Jackson[2]の「意味と文」の関係を参考にし、記述文の持つ意味の構造を以下のように捉えた。

(1) 記述が文で表わされる場合、意味の基本構造は、状況の型(動作)、関与者、周囲、から成る。これをここではVPC組と呼ぶ。VPC組では、関与者や周囲の動作に対する関係を明示すると、1) 文法に従った表現では厳密さが要求されるような要素の位置を固定する必要がなくなり、2) 一つの動作に対し、複数の関与者や周囲が表わされるようになる。

簡単な文では、関与者と周囲が共通で、動作が2~3個使われる場合がある。また、PAやCRの部分がand/orで結ばれることもある。VPC組をandやorで結んで等位節の複文が構成されることもある。

(2) 関与者や周囲が句で表わされる場合、修飾語としては、冠詞、形容詞句、ofで始まる句、関係代名詞や動名詞等のVPC組による修飾がある。周囲では、前置詞句で表わされ、関与者では名詞句となる。

関与者や周囲の句に節を用いた修飾がある時は、文全体として見ると、VPC組が再帰的に用いられる。

(3) 関与者や周囲が節で表わされることがある。これには、周囲では、to不定詞や従属節が用いられる。関与者では、that節や疑問詞で始まる節がある。何れもVPC組として表現され、文全体として見ると、VPC組が再帰的に用いられる。

(4) 関与者として表わされた記述は、元来、対象の性質について述べたものだから、この部分の記述は属性機能の記述と見なすことができ

る。

属性記述の面から見ると、この性質を持つ対象ならば、何でも表わし得る。H. Jackson[2]の意味分類において、関与者の役割としてAgentiveが含まれている。この点からも、関与者として動作の対象に限らず、動作主体の属性に関してもこの記述を用いることができる。そこで表記法には、冗長性を許し、動作主体の属性の記述と、動作の対象(客体)の記述の区別を含める。

(5) "is_a"で表わされる性質は、状況の型をstateとしてVPC組で表わし得る。しかし、これまで提案された"is_a"で表わされる性質は、属性を表わすものとして(type, value)やtype = value等で表現されてきた。そこでこの形はVPC組の節表記ではなく、句の表記の拡張として、表すことにする。等号以外の関係、例えば、time > 5msec等も表わしたい。

(6) 環境記述の側面では、環境の種類の名前とその用途や役割の組による記述が必要になる。

環境の記述は対象の周りの属性を表すので、属性記述と同様な記述ができるが、ここでは簡単化して通常の属性の名前と同様な表記とした。

(7) 識別子や使用法の側面は、環境が決まると一意になるようなことが分かる構造が必要になる。

このような捉え方に基づき、記述表現法を以下の考え方で構成した。その結果は、図2に示される。図では[・]は省略可能な要素、{・}は、まとめて一つであることを表す。要素を並べた時は、andで成立、|で並べた時はorで成立することを表す。

(1)の前半は、VPC組の表記構造であり、これは、状況の型ST、関与者PA、周囲CRか

```

SF ::= PF | PA | id | us | Env | SF SF | { SF/SF }

clause ::= {( ST [PA] [CR] )}

PF ::= clause | clause clause | {clause|clause}

ST ::= v |{v v}| {v|v}
Partname ::= Affected:|Agentive:|Recipient:|Resultant:|Attribute:|
             Eventive:|Exeternal Causer:|Instrument:|Positioner:
Circname ::= Locative:|Temporal:|process:|
             Respect:|Contingency:|Degree:
Prep ::= in:|at:|between:|from:|to:
Al ::= a | the | | one
Ph ::= noun | ( noun Value ) | (Ph Ph) |{ Ph | Ph }

Value ::= string | number | op(value)
op ::= < | = | >
Md ::= adjective phrase | ( Md Md ) | ( Md|Md )

PA ::= p([Al][Partname] Ph [Attr:Md] [of:Ph] [(Partname clause)] ) |
      P([Al][Agentive:] Ph [Attr:Md] [of:Ph] [(Partname clause)] ) |
      p(Partname clause) | P([Agentive:] clause) | PA PA | { PA | PA }

CR ::= c([Circname][Prep] Ph [Attr:Md] [of:Ph] [(Partname clause)] ) |
      c( Circname[Prep] clause ) | CR CR |{ CR | CR }|

Env ::= e( noun, Value ) | Env Env
Cmd ::= command | command-line
id( Env: Ph )
us( Env: Cmd )

```

SFの精密化

```

SFe ::= PF | PA | id | us | Env
SFa ::= SFe SFe | SFe SFa | SFa SFe
SFo ::= {SFe | SFe} | {SFe | SFo} | {SFo | SFe}
SFc ::= {( SFa ) | SFe} | ( SFo ) SFe | {SFe | (SFa )} | SFe ( SFo )
        | ( SFc SFc ) | ( SFc | SFc )
SF ::= SFe | SFa | SFo | SFc

```

図 2: A proposed notation of descriptive names

らなる clause で示される。関与者や周囲の状況の型に対する役割は Partname と Circname で表わした。(1) の後半の構造は、ST、PA、CR をそれぞれ and や or で繋いだ表現となる。

(2) の前半は、PA や CR の修飾語の表現とした。句の中心は Ph で表わされる。CR の前置詞句では前置詞が必要な場合も考慮して、Prep を設けた。(2) の後半は、PA や CR の中で節として Ph を修飾する場合である。節の修飾では、Ph の clause に対する役割を Partname で示した。

(3) は、PA や CR が節だけ述べられている構造として加えた。

(4) については、提供機能 PF は clause やその組合せにより表わされる。PA は、属性機能の記述を表わす。動作の主体を表わす場合は関数記号を大文字の P で、動作の客体（関与者）には小文字の p で表わして役割を区別した。ここで、P は、Partname = Agentive となる、p の特別な場合である。冗長だが便利さのため導入した。

(5) に対しては、Ph に、"is_a" の表現を含めた。Ph に含まれる "is_a" である (type, value) の表現は、通常は等号の関係を表わす。value を拡張して、op を用いて等号の関係だけでなく、不等号の関係も表わせるようにした。

(6) や (7) に対しては、識別的記述側面の表記には、一意に識別するために環境の値を加えた。使用法は us、識別子は id、の関数記号で表わした。環境的記述側面は、Env で表記した。

記述名は 5 つの側面から成る。これらの側面は単独で記述してもよいし、and や or で組み合わせてもよい。このような記述名は図 2 の中の SF として表記される。

2.2 記述表記法の解釈

図 2 のような構造の意味を以下のように解釈する。

(1) 同じレベルの括弧の中の VPC 組は、and の関係と見て、PA と CR は ST の説明をしていると解釈する。PA のうち、p は動作の働きかける他の対象、P は動作主を表わす。

VPC 組が再帰的に用いられた場合（括弧のレベルが変わる）外側の Ph や ST を修飾する節と解釈する。

(2) 単一要素からなる SF が and で並べられた時は、両要素の意味が成り立つと解釈する。結果として、矛盾が生ずる場合は、満たす対象は無い、と読む。以下に例を挙げた。単一要素からなる SF が or で並べられた時は、どれかが成り立つと解釈する。

VPC 組と us が and で並べられていれば、VPC 組として示される提供機能に対応する使用法は、us で示される環境では us の Cmd の値と解釈する。PA と id が並べられている時も同様に解釈する。

id と us が異なる Env で表されている時は、それぞれ異なる環境の値と解釈する。

(3) (2) の関係を利用すると、記述表現で問合せが表現可能となる。記述は関数記号と括弧を用いて表される。記述内容が分かっている場合は、関数記号の後ろの括弧の中に記述がある。括弧の中が空な場合、括弧の中の値を求める表現と解釈する。こうすると、問合せで求めたい記述側面は、括弧の中が空な表現を用いることで実現できる。

3 内部構造の構成法

3.1 構成上の考慮点

内部構造では、1) 利用者向き表現に対応させた構造や、2) 名前管理、特に、記述名管理では、利用者からの問合せに基づいた名前解決処理が重要となるので、これに向けた構造が必要になる。

1) に対しては、利用者向き記述側面の各々に対応した構造を持ち、記述の多線性に対処できることが挙げられる。ここで記述の多線性は、

構文 (syntax) としては、利用者向き表記法から導かれる項目として、いろいろな記述側面の記述ができ、また、必要な部分の記述だけが述べられ、不要な記述の省略が可能であること、記述順序は自由でよいこと、とする。意味 (semantics) としては、記述の質、抽象性、具体性、に対処できることとする。

2) に対しては、名前解決で重要となる問いの記述側面 (複数の可能性あり) から答の記述側面に簡単に到達できるような構造が必要になる。

このことから、記述側面毎の構造、記述側面間の写像について以下で考察する。

3.2 利用者表現法に対応した記述側面の構成

一つの対象を表す記述側面は図 2 では、PA, PF, id, us, Env として表される。各々の構造を 1) Md や Ph では、and/or の接続で複数の要素が並ぶ可能性があること、2) clause では clause の要素を他の clause が修飾することがあり、構造として、再帰的に用いられる可能性があること、3) PA や CR では、句または節が単独で構成されることがある点を考慮して、構成する。即ち (括弧内の値を図 3 中の値として)、一つの対象を表す使用法と対象の識別子の記述側面は、環境の値と共に構成する (Id, Us)。提供機能では、VPC 組が基本となる (vpc)。関与者や主体の属性は属性機能として表される (PA)。周囲 (CR) の要素は、句か節の何れかからなる (phcl)。関与者や周囲では、記述が句 (静的属性) を表す場合は、鍵となる意味要素は Ph の値である。Ph の限定子により属性の意味が変化し、この記述の表すもの (使用法、コマンド) が変わる。同じ Ph で修飾語の違いで、使用法の区別のつく最小の修飾語 (限定子) を Ph の下に並べる。続いて環境を通して、使用法への写像を行なう。一例を図 4 に示した。関与者や周囲の記述が節の場合は、役割の下に使用法が区別できる最も簡単な記述を置き、ここから環境を通して、使用法への写像を行なう。

記述の種類、記述要素の有無、使用個数等の表示を行なうことで多様な記述に対処した。

以上を纏めると、図 3 となる。

3.3 記述側面間の関係

一つの対象の記述側面に注目すると、属性的記述側面は対象自身の値であり、識別子や使用法は環境に依存して値が定まる。属性的記述側面から識別的記述側面への写像は

environment(descriptions) → identifier or usage

と表される。

3.4 記述の多様性を考慮した構成

3.4.1 多様な表現を支援するための属性的記述側面の構造

利用者表現法は記述文を意味に基づいて簡単な構造を入れたものであるから、記述の内部構造も意味に基づいて捉え、分類するのが自然と考えられる。そこで、記述の意味の要素について、関連するものを纏め、共通化した属性をタイプで表す。記述の抽象性と記述の量に基づいて、木構造に似た構成を作る。

提供機能の記述では、基本的には VPC 組で構成される。この要素の中で、共通性が高いのは、動作なので、同じ動詞を中心に纏める。関与者 (PA) や周囲 (CR) で共通性が高い記述要素は、関与者や周囲の役割 (図 2 中、Partname や Circname の値) であるので、これを並べ、この下に役割に対応した属性の記述を書く。関与者や周囲では、記述が句 (静的属性) を表す場合は、鍵となる意味要素は Ph の値である。Ph の限定子により属性の意味が変化し、この記述の表すもの (使用法、コマンド) が変わる。同じ Ph で修飾語の違いで、使用法の区別のつく最小の修飾語 (限定子) を Ph の下に並べる。続いて環境を通して、使用法への写像を行なう。一例を図 4 に示した。関与者や周囲の記述が節の場合は、役割の下に使用法が区別できる最も簡単な記述を置き、ここから環境を通して、使用法への写像を行なう。

```

struct stelm {
    char term[Chnum ] ;
};

struct ST { /* verb structure */
    int num;
    int op[Stnum - 1];
    /* operations between elements */
    struct stelm sttab[Stnum];
};

struct vpc { /* a VPC group */
    unsigned int form;
    /* indicating using pattern,
        St, Pa, Ps or Cr */
    struct ST *pt_st;
    struct PA *pt_participant;
    struct PA *pt_subject;
    struct CR *pt_circum;
};

struct role_vpc {
    int role;
    struct vpc vpc_cont;
};

struct of_mod { /* of-phrase structure */
    int num;
    struct ph_elements *pt_ph;
};

struct mod { /* modifiers */
    unsigned int form;
    /* using indication
        of < Art ofp Adj cla > */
    char *pato;
    struct of_mod *pt_of;
    struct Md *pt_md;
    struct role_vpc *pt_vpc;
};

struct Md { /* adjective phrase */
    int num;
    int op[Mdnum - 1];
    struct stelm mdtab[Mdnum];
};

struct ph_element {
    int form;
    /* Noun, Tuple, or Tuple
        with operation */
    struct stelm sttab[2];
    int op; /* Tuple with operation */
};

struct ph_elements {
    int num;
    int op[Phnum -1];
    struct ph_element phtab[Phnum];
};

struct ph_struct { /* phrase */
    unsigned int form;
    /* indication of usage
        < phr cla Rl Mdf >*/
    int role; /* role description */
    struct mod *pt_mod;
    struct ph_elements *pt_ph;
};

union phcl {
    struct ph_struct *pt_ph;
    struct role_vpc; *pt_vpc;
};

struct PA { /* attribute description */
    int num;
    int op[Pnum - 1];
    int form[Pnum];
    /* phcl use: T = phrase, F = clause */
    union phcl pa[Pnum];
};

struct Cir {
    unsigned int form;
    /* indication of usage,
        < pre Mdf Rl phr cla >*/
    int role;
    struct stelm pretab; /* preposition */
    union phcl cirterm;
};

```

```

struct CR { /* circumstance */
    int num;
    int op[Cnum - 1];
    int form[Cnum];
    struct Cir cir[Cnum];
};

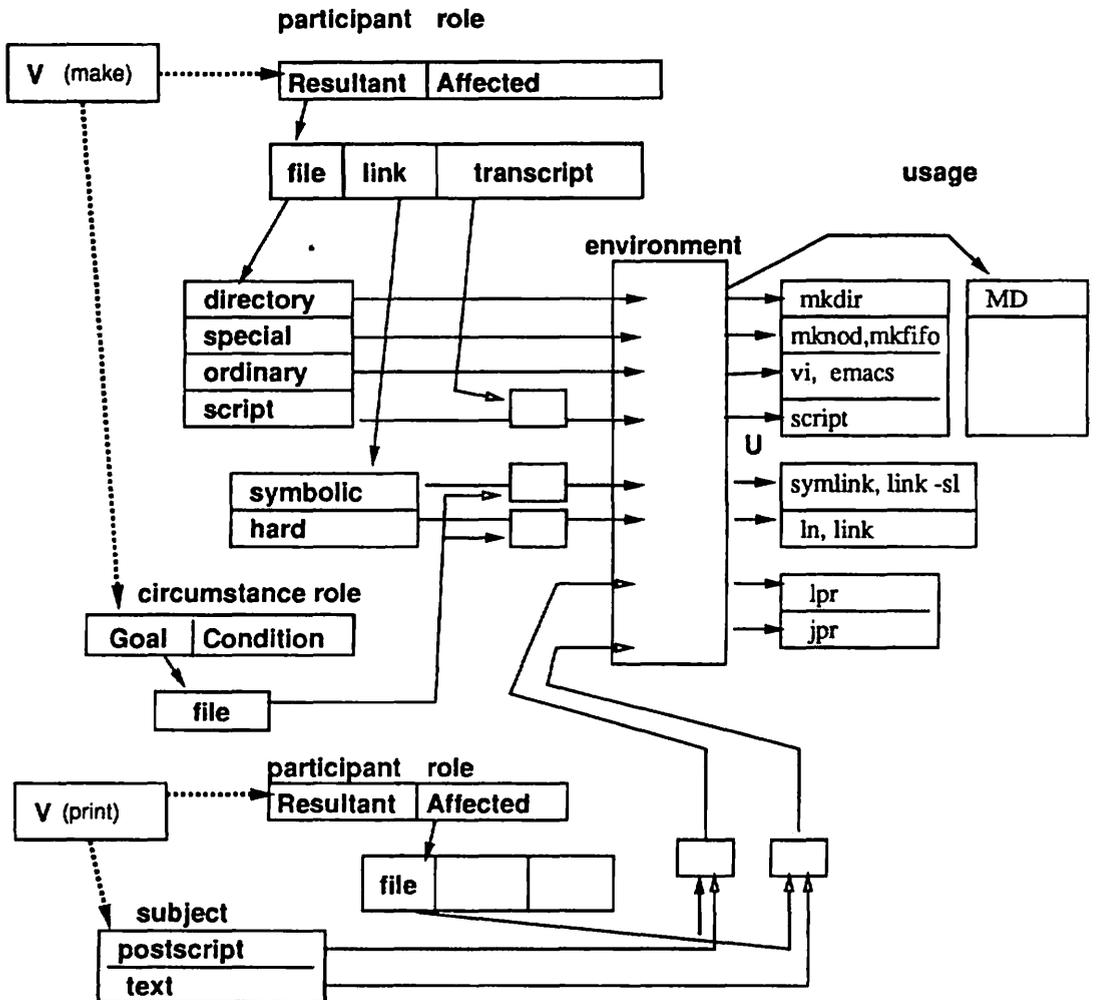
struct Env {
    int num;
    int op[Enum - 1];
    struct ph_elements *env[Enum];
};

struct Id {
    struct Env env;
    struct ph_elements *id;
};

struct Us {
    struct Env env;
    struct stelm *cmd;
};

```

図 3: 一つのオブジェクトの記述側面の構成 (C 言語風)



☒ 4: Mapping structures from attribute aspects to distinguishing ones

属性機能は主に、図2中、PAと表される。ここで、最も簡単に表記した場合は、句の場合は、Phだけになる。Phが(noun Value)の形でValueが数値の場合は、具体値と見なす。その他の場合は、Phの値の抽象度により、木構造が構成される。この構成要素には修飾語として、Attr、of-Ph等の静的な属性、VPC組に対応する動的属性があるので、これらを記述の抽象性に沿って並べる。

3.4.2 多様性実現のための写像構成

属性的記述側面の記述の詳しさは、「これだ」という形で一意に定まることはない。記述要素を組み合わせることで対応する使用法や識別子が表される。記述の仕方は一通りではないので、冗長な組合せの記述も起こり得る。この記述の中で、使用法や識別子を識別する最低限(μ)の記述要素の組合せの記述が見つければ、この間で写像を張ることとする。

このことで、識別的記述側面に写像を持つ記述より詳しい記述は、すべて、同じ、写像で、識別的記述側面に対応させることができる。

提供機能の記述の例では、図4に示されるように、動作がprintの場合、印刷を行なう主体(通常はプリンタ)が、Postscriptの属性を持つ場合は、環境Uでは、lprコマンドになり、主体がtextの属性の時は、同じ環境で、jprに写像されることを示している。これより粗い記述の場合は環境Uに対しては対応コマンドはない。また、ここに記述されているよりも多い記述要素を持つ場合もこれらの写像関係は有効に働く。

3.4.3 環境と識別的記述側面との関係

識別的記述側面の考察範囲は、利用者向きで「計算機」に関する資源、特に、各種OSでの資源の識別と使用法に限っているので、環境の種類は、(暫定的に)場所(location)、計算機の種類、OS、計算機利用の立場(視点、用途)、だけを考える。

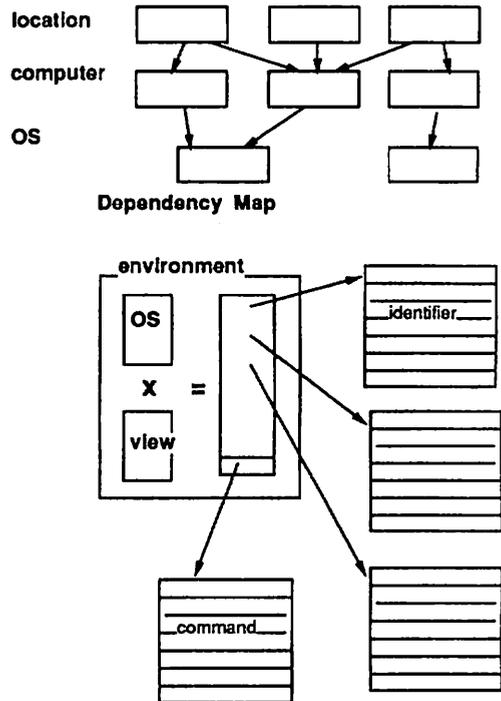


図5: Map structures between environment-descriptions, identifiers and usages

これらを記述名に係わる環境として見ると、OSに関する識別的記述側面(使用法、識別子)は環境を場所と指定しても、計算機と指定してもOSと指定した場合と同じ値になると見なせるので、場所、計算機、OSには前者から後者へ従属関係を入れる。OSの中では、用途(view)によって識別法が異なるので、この関係も考慮する。用途としては、equipment、file、commandを考える。前二者は識別子として、後者は使用法としての識別に対応する。

以上を纏めると、環境としては、OSの種類を基本とする。OSの種類と用途で決まる環境値に応じて、識別子や使用法を表す文字列配列への写像を構成する。他の環境値は、OSの種類への写像を行なう(図5)。

このことにより、利用者の環境指定が、直接、OS名でなくとも、対処できるようになる。

4 考察

4.1 内部構造の構成

記述名の名前管理では、利用者からの問合せ等の記述を名前解決して、必要な答としての記述を返す処理を行なう。このために、利用者表現法の直接の写像としての個々の記述の内部表現と、管理用情報ベースとの値を比較する処理が必要になる。この理由から、記述名の内部構造を、記述文の持つ意味構造に基づいた利用者向き記述表現法に直接対応する構造と、集合としての構造に分けた。

利用者向き記述表現法に直接対応する構造では、記述の各要素に対応させて、構造を作り、使用の有無の標識をつけることで、記述の多様性に対処できるようにした。問題点としては、記述文の意味の構造から構成したので、複雑な構成になっている点が挙げられる。現実問題としてこのような複雑な記述が行なわれるかを調べてみる必要がある。

管理用情報ベースでは、一つの記述側面の要素が分解されるので、個々の記述では、個別に書く必要のあった共通性の高い要素は一つになり節約できる反面、個々の記述では、必要のなかった記述要素間にポインタを用いて関係付ける必要が生ずる。記述要素が表の形になっているので、参照頻度により、並べる順序を変えることも必要になる。この点は、実験システムを構成する時の課題の一つになる。

4.2 記述側面間の関係付けの問題

属性的記述側面で、環境対応に異なる識別子や使用方法を持つ場合は、同じ性質を持つ対象が(異なる環境に)複数ある場合である。記述の詳しさは一通りには決まらない。属性的記述側面を細かく記述すると異なる対象間の性質の共通性が薄れ、各々が別個の属性を持つことになり、対応する識別子や使用方法は一つの環境でしか持たなくなる。すると、3.4.3節で構成した写像は一つの環境の所だけが値を持ち、他の環境からの写像の行った先の値は空になる。この問題

に対しては、属性的記述側面はできるだけ共通な性質を述べることにし、細かな記述表現にはローカルな表を構成し、そこで調べることが考えられる。

利用者向き表現法では記述できる構造を持っているが、3.4節では、共通化を重視したため、属性的記述側面からの写像の結果が、使用法を組み合わせた記述になるような場合は考察されていなかった。例えば、「あるディレクトリの下で或る処理をする」というような条件つき記述では、「ディレクトリを決定する処理と、或る処理」との異なるコマンドが複数組み合わせられることになる。このような個別化処理の実現法も問題となる。

この点は今後の課題として残される。

5 おわりに

本論文では、利用者表現法に対応した記述名の内部構造の一案を示した。

内部構造には、記述文の持つ意味構造に基づいた利用者向き記述表現法に直接対応する構造と、記述側面を意味関係を利用して共通化して管理用情報ベースの二種類の構造を持たせた。これらの構成法について考察した。

参考文献

- [1] Kouda, F. : A new notation of user-friendly descriptive names, *ICOIN-9, to appear (1994)*.
- [2] Jackson, H. : *Grammar and Meaning*, Longman, p.292 (1990).