

4F-2

分散名前管理における記述名の管理方式

古宇田 フミ子 田中 英彦
(東京大学 工学部)

1. はじめに

分散環境では正確な名前が分からない可能性が単一システムより大きいであろう。計算機資源の利用時に、識別子以外の名前、例えば、自然言語に近い表現で資源の機能や使い方を示し、これにより、資源利用を可能にする手段を提供することも必要となる。

筆者は資源の機能、性格等を説明するものを記述名とし、資源の機能や他の資源との関係を記述可能なKL-ONE [1] を道具に用いて、資源分類の見直しを行っている。この分類により、記述名の構成を考察してきた [2] が、今回は、管理面からこの分類と記述名との関係を述べる。

2. 前提事項

2.1 KL-ONE

KL-ONEは必要性(常に成り立つもの)と状況性(成立することもしないこともあるもの)を各々記述部、宣言部に分け、内部構造を継承する意味ネットワーク(SINet)である。記述部では対象の生来的属性を表し、宣言部で対象の一時的な述語関係を表現できる。この意味ネットワークではデフォルトや例外を認めず、上下方向は包含関係になっている。筆者は、KL-ONEに基づく分類に基づいて記述名の構成を行うと、分散環境における分散透明性を実現できると考え、これまでにその分類の考察を行ってきた [2]。

KL-ONEに関する主な用語は、概念(concept)、役割(role)、補概念(filler)、一般概念(generic concept)、個別概念(individual concept)等である。役割間の関係を示す構造として、役割値写像(RoleValueMap)と、構造化記述(StructuralDescription)とがある。

SINetの節は概念であり、節にはこの概念を説明する役割や役割の対象となる概念(補概念)、役割間の関係構造がある。図1.に一例を示した。file access right という概念はclass とcapabilityという役割を持ち、その補概念はそれぞれ user と access mode である。この意味は FileAccessRight has Class which is User. である。class とcapabilityという役割の間にはaccess checkという構造記述があることも示している。file access right という概念はfileという概念に対して、スーパー概念でありfile概念はfile access right 概念のサブ概念である。

2.2 SINetと識別名、記述名との関係付け

これまでの分類の考察により、SINet においては、概念

はfile access right やprocessing等のように抽象的な性質を示す抽象概念と、存在するもの(例、file)を指す実体概念が観察される。実体概念のSINetの節間の包含関係を幹として捉え、この葉の部分の個別概念に識別名を付けて、対応させる。抽象概念は実体の性質を示すのに使用されるため個別概念は存在しない。実体概念の幹は何木かできる(*)が、これを資源分類用で使用可能である。記述名は役割と補概念や役割間の構造を用いて表現できる。記述名は対象の生来的関係のみとし、分散の差が表れる概念の部分は名前管理に含める構成法により、分散透明性を実現できる形が可能である、ことが分かった。

3. 記述名管理方式の設計

3.1 SINetの名前管理への設計方針

SINetは包含関係になっているので上位の節で加わった役割は必ず下位まで継承される。これを実装する場合、包含関係をそのままポインタでつなぐと継承の関係が見えにくい。役割の構造は下位に行くほど増えるので構成が複雑になる。SINetの各節の概念に注目すると、役割が新たに追加されている所があったり、サブ概念は或る概念の役割を制限することにより生じたり、複数の概念を継承してできていることが分かる。一方、記述名の要素は、役割と補概念や役割間の関係の表現として捉えた。

そこで、SINetから記述名を導き出す方策として、継承と役割の関係が明示的に表わされ、記述量も減らせるように、実体概念の節を中心として、SINetの構成を図2.のように分割した。即ち、一般概念の場合(図2.a)は、一つ

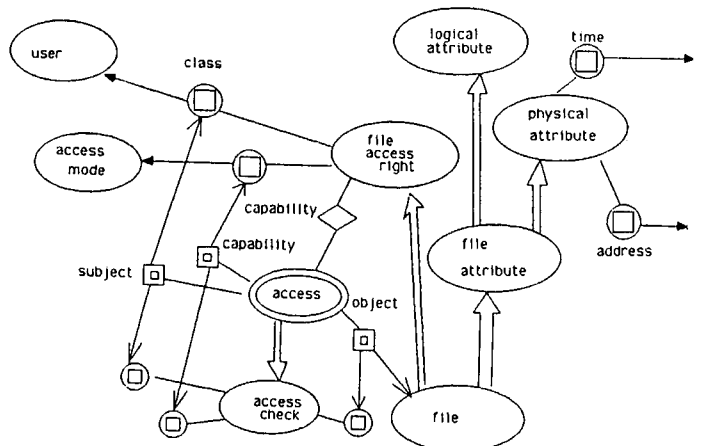


Fig. 1 Example of SINet written in KL-ONE

の概念に対し、i)ここで新たに加わった役割やその関係（RVM、SP）の記述、ii)スーパー概念へのポインタと制限する基となる役割及びこの概念における役割の補概念、iii)この概念のサブ概念へのポインタと、サブ概念を作る基となった役割、の三要素のみを書き、概念間の関係が明白にできるように単位に分割した。

一方、個別概念の場合（図2.b）はサブ概念がないこと、個の集合を示す必要があることから、i)スーパー概念へのポインタとこの個別概念を作る基となった制限役割、ii)個別概念の枠の記述、個別概念に対応する識別名へのポインタを一単位として分割した。

3.2 記述名から識別名の導出

SINet では、探索の入口をどこにするかが問題となるが、2.2の(*)を利用し、幹毎に複数の入口を設ける。入口は実体概念の最上位とする。記述名の形式は

（幹の名）{ { role = filler } , RVM, SP }

となる。探索方法は 3.1の分割構成法を利用し、提示された記述名の役割と、注目する節の概念に新たに加わった役割、補概念や役割間の構造をとを比較し、関係するサブ概念に進む。これを繰り返す、個別概念の所まで進み、識別名を導き出す。識別名が一つにならない時は、この記述名は集合を表している。

3.3 識別名から記述名の導出

識別名が与えられた時、先ず、これに対応する個別概念の枠に進む（図2.bの矢印）。ここで満たしている役割や補概念があれば、記述名の一部として書き出し、関係するスーパー概念に進む。スーパー概念は同じレベルの記述名の一

部を取り出すため breadth-firstで調べる。この概念で新たに定義された役割や補概念、その構造を記述名の一部に書き加える。これを繰り返す、最上位（抽象概念でスーパー概念を持たない。）に来たら終了し、書き出したもの全体が記述名となる。

4. 考察

4.1 SINetと節の概念単位の分割

本来は、概念の持つ役割はすべてその概念の所で記述する必要があるので、SINet の下位に進む程役割の記述が長くなる。筆者の方法はSINet は包含関係であり、例外やデフォルトを認めないことを利用して、役割の記述を、役割が新たに加わった所だけにし、言わば、差分表現を採ることにより、同じ記述は繰り返さず冗長性を取り除いている。

SINet を辿る場合は上位から下位へと、下位から上位への両方向のポインタが必要である。節の概念単位で分割しない場合も上位方向と下位方向の両方が必要なので、分割した場合と比べてポインタ数の増減はない。

役割と継承の関係、役割の構造の関係に注目してSINet を分割したので記述名から識別名への道を辿る場合、記述名の中の役割と節概念の役割の一致の検索が速く、又、次段階のサブ概念を迅速に求めることができる（図2.a）。

スーパー概念に対しては役割と補概念をつけ、サブ概念には役割のみを付す分割法は非対称であるが、上位方向と下位方向とは探索するものが異なる、検索に冗長な要素をつけない等の理由でこのようにした。

4.2 検討項目

3.章の構成法では検索プログラムはループを用いた繰り返しが多いので、比較的小さくできたが、分割したディレクトリの一単位の構造は役割の記述については差分構成ではあるが、単純ではない、所要記憶容量の問題を検討する必要がある。分散処理に適用する場合にこのディレクトリをどこに配置するか、分散環境の利用者に使い易いか、も検討する必要がある。

5. おわりに

KL-ONEに基づくSINetを記述名構成に利用し、これをディレクトリとして構成する設計法と、これに基づく、識別名、記述名間の導き方の一案を示した。

現在は、この構成をCを用いて、プログラムに書き下している。

今後の課題は、このシステムの適応範囲の評価及び分散処理への応用、と他方式との比較である。

参考文献

- [1] PROCEEDINGS OF THE 1981 KL-ONE WORKSHOP, Fairchild Technical Report No.618, FLAIR Technical Report No.4 (1982)
- [2] 古宇田、田中：分散名前管理のための記述名の利用、情報処理学会第35回全国大会, 30-4 (1987)

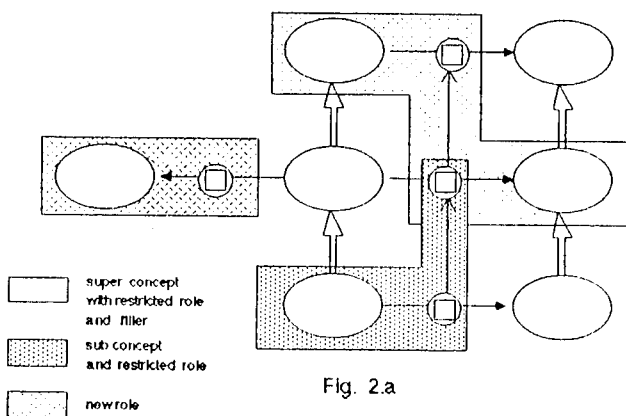


Fig. 2.a

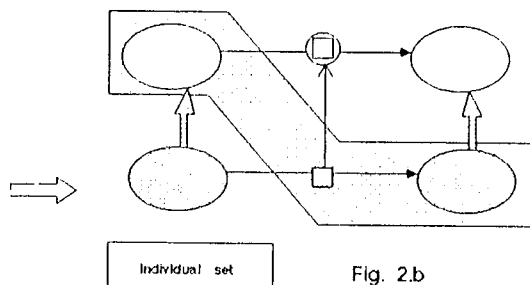


Fig. 2.b