

インタードメインルーティングプロトコル (IDRP)のためのポリシー記述言語

村田 顕宏 塚本 昌彦 西尾 章治郎

大阪大学工学部情報システム工学科

インタードメインルーティングプロトコル (IDRP) は、OSI やインターネットでの普及が見込まれている政策的経路制御プロトコルである。IDRP ではネットワークの運営や管理についてのローカルな方針 (ポリシー) を反映した緻密な経路制御を行えるが、反映すべきポリシーが多くなると極端に設定が複雑になる。そこで本研究では、経路制御に関するポリシーと設定に必要なさまざまな知識を記述する言語を設計し、この言語によって記述された知識とポリシーから適切な設定を行うツールを実装した。設定に必要な知識はすべて is-a 関係で統一的に表し、推移律や継承関係に基づく推論機構を設けた。また、ルール形式による抽象的なポリシーの記述を可能とした。

A Policy Description Language for Inter-Domain Routing Protocol (IDRP)

Akihiro MURATA Masahiko TSUKAMOTO Shojiro NISHIO

Dept. of Information Systems Engineering, Faculty of Engineering, Osaka University

Inter-Domain Routing Protocol (IDRP) is a standard policy routing protocol of OSI as well as Internet. Using IDRP, an administrator of a domain can provide a configuration description regarding local policies of management and control of the network. However, such descriptions generally become very complex when the administrator wishes to handle many policies in detail. In this paper, we propose a policy description language to describe routing policies in an abstract and flexible manner; pieces of knowledge concerning the objective domain are described as an is-a relation and several associated rules. We also demonstrate the tool which we have implemented to generate routing configuration of IDRP from the policies described by this language.

1 はじめに

インタードメインルーティングプロトコル (IDRP) [5] [10] は、OSI の CLNP (Connectionless Network Protocol) ネットワークにおいて政策的経路制御を行うためのプロトコルである。現在はインターネットにおいても経路制御プロトコルとして IDRP を用いる方向で研究が進められており、将来標準的な政策的経路制御プロトコルとなる可能性が高い。

IDRP では、インターネット網をドメイン (domain) という単位に分け、経路制御を階層化することによって網規模の拡大に対応している。また、ドメイン間の経路制御に各ドメイン内のネットワーク資源についての運用方針 (ポリシー) を反映させることができる。ここで、経路制御に反映されるポリシーとは、例えば

- ドメイン A を通る経路は避けたい。
- ドメイン B へのパケットにはネットワーク資源を使わせたくない (中継したくない)。

というような、経路の選択やパケットの中継などについての方針である。

IDRP では、各ドメインのポリシーは経路の選択や経路情報の配布に関する細かい設定を通じて経路制御に反映される。このため、単純なポリシーを反映させるためにも、一般に、煩雑な設定を行わなければならない。また、これまでに提案されている設定方法 [10] では、アドレスやネットワークポロジ、個々のポリシーと、設定事項との関係がわかりにくいいため、ポリシーが複雑になるにつれて保守作業が困難になり、ネットワーク管理者の負担が極端に増大するという問題がある。

このような問題は、ネットワーク管理者がアドレスやネットワークポロジ、組織間の関係などの知識からポリシーに沿った経路制御の設定を行う過程を、自動化することで解決できる。そこで本研究では、経路制御の設定に必要なさまざまな知識とポリシーを記述する言語を設計し、この言語を用いて記述された知識とポリシーから適切な設定を行うツールを実装した。

今回開発した言語では、設定に必要な知識はすべて is-a 関係の知識として知識ベースに記述する。ポリシーは知識ベースで定義されたドメイン集合に対して記述する。また、知識ベースから具体的なポリシーを導出するためのルールとして、より抽象的なポリシーを記述できる。これによって、経路制御に関するポリシーをアドレスやネットワークポロジなどから独立した形で記述することが可能となり、ネットワークポロジや組織間の関係、組織の構造などの変化による更新作業を簡素化することができる。自動設定ツールでは、is-a 関係の推移律や継承関係に基づく推論機構を設けることで、記述された知識からより多くの知識を導出できるようにした。また、より限定された対象に対して記述されたポリシーを優先することで、デフォルトや例外の記述を容易にした。

本稿では、まず 2 章で IDRP の概要を述べる。3 章では本研究で設計したポリシー記述言語の設計方針と構文について述べる。4 章では経路制御自動設定ツールの実装について述べる。5 章では、言語の表現力とツールの性能について考察する。最後に 6 章で、本研究のまとめと今後の課題について述べる。

2 IDRP の概要

IDRP は、OSI の CLNP (Connectionless Network Protocol) ネットワークにおいて政策的経路制御を行うためのプロトコルであり、ISO (International Organization for Standardization) の国際標準原案として仕様が決まっている。もともと IDRP は、インターネットにおける経路制御プロトコルのひとつである Border Gateway Protocol (BGP) [3] を ISO の OSI 環境に適合させ、そこからさらに発展させたものだが、現在インターネットにおいても、BGP の研究を凍結し、経路制御プロトコルとして IDRP を用いる方向で研究が進められており、将来標準的な政策的経路制御プロトコルとなる可能性が高い。

IDRP では、インターネット網をドメイン (domain) という単位に分ける。ドメインとは、共通の運用方針に従う物理的に連続なネットワークの集合である。IDRP はドメイン間の経路制御を行ない、ドメイン内ではそれぞれ独自の経路制御プロトコルを用いることができる。このような経路制御の階層化によってネットワークの大規模化に対応でき、また、既存のネットワークとの接続を容易にすることができる。

IDRP の特徴を以下に列挙する。

- path-vector プロトコルである。
- 通過トラフィックに対して、その送信元や受信先のドメインに応じてネットワーク資源の使用を制限できる。
- ルーティング ドメイン コンフェデレーション (Routing Domain Confederation:RDC) によ

て、複数のドメインをグループ化し、経路情報を削減できる。

- 識別属性 (distinguishing path attribute) によって、一つの目的地に対して複数の経路をサポートすることができる。
- 複数の経路情報を一つに集約 (aggregate) することによって、経路情報を削減することができる。
- 経路長 (ステップ数) だけでなく、伝送料金、信頼性、または伝送速度を考慮した経路選択ができる。

2.1 IDRP のポリシー設定言語

ISO の IDRP に関する規約 [10] には付録としてポリシー設定言語の構文の例が挙げられている。本研究で実装する経路制御自動設定ツールでは、この構文の設定用言語があらかじめシステム側で供給されているものと仮定する。

このポリシー設定言語は三種類の文で構成される。すなわち、経路選択に関するポリシーを表す Preference 文、経路情報の配布に関するポリシーを表す Distribution 文、そして経路の集約に関するポリシーを表す Aggregation 文である。

各文はテンプレートと式を持つ。隣接ドメインから得られた経路情報は順次 Preference 文のテンプレートと照合される。マッチしたのに対しては式が評価され、それによって優先度が付けられる。この優先度によって、各目的地への経路が選択される。

Distribution 文のテンプレートにマッチした経路情報は式によって配布先を制限される。隣接していないドメインに対しても配布を禁じることができる。

Aggregation 文のテンプレートにマッチした経路情報は一つに集約され、隣接ドメインに配布される。

このポリシー設定言語の記述例をリスト 1 に示す。

```
PREF{XX:WW:0:*}/.*XX:WW:2201.*/  
  (qos_none)=200-hopcount();  
PREF{XX:WW:0:*}/.*XX:WW:2201.*/  
  (error)=200-error();  
PREF{XX:WW:0:*}/.*XX:WW:2201.*/  
  (expense)=100-expense();  
PREF{XX:WW:0:*}/.*XX:WW:2201.*/  
  (delay)=100-delay();  
PREF{nlri_any}(qos_none)=100-hopcount();  
PREF{nlri_any}(error)=100-error();  
PREF{nlri_any}(expense)=-expense();  
PREF{nlri_any}(delay)=-delay();  
  
AGGR{XX:WW:*}={rdi XX:ZZ:1024} man;  
  
DIST{XX:YY:*}={internal_bis}select_only;  
DIST{nlri_any}=select_on  
  {prohibit_dist(XX:YY:000b,XX:YY:102a);};
```

リスト 1: IDRP の設定例

3 経路制御ポリシー記述言語の仕様

本研究では、次のような方針に従って、中継ポリシー、経路選択ポリシー、経路集約ポリシー、およびIDRPの設定に必要な知識を記述することができる経路制御ポリシー記述言語を設計した。

- is-a 関係式による知識の表現
- ドメイン集合によるポリシーの抽象化
- 直観的に分かりやすい構文
- 対象の広さによるポリシーの優先順位付け
- ルールによるポリシーの抽象化

以下では、設計した言語の構文と用法について概説する。

3.1 知識ベースの構文

ドメインのアドレスやドメイン識別子 (Routing Domain Identifier: RDI), 隣接ドメイン, 組織の構造や組織間の関係など, 経路制御の設定に必要な知識はすべて is-a 関係式で明示されていなければならない。

例えば、「A社はB社のライバル企業である」「ドメインCとドメインBの間の回線は質が悪い」といった知識は次のように記述する。

```
company_A is_a competitor of company_B.  
rd_C-rd_B is_a bad_circuit.
```

また、あるオブジェクトが他のオブジェクトの集合である場合、次のように集合のメンバをまとめて記述できる。

```
competitor of company_B is_a  
[company_A, company_C]. %略式メンバ定義
```

本稿では型と実体を区別せず、ともにオブジェクトと呼ぶ。一般に、is-a 関係とは型に対して定義され、その型に属する実体の集合の包含関係を表すが、本研究ではこのような包含関係のほかに、集合とその要素の関係を含む拡張的な関係を示すものとして扱う。

オブジェクトそのものか、「オブジェクト of 属性名」という構文で仮想的に表現されたオブジェクトの属性値を、本稿では属性式と呼ぶ。属性値もオブジェクトである。

本研究で用いる is-a 関係において、次のような推論規則を導入する。

継承: 任意の属性式 X, Y, 任意の属性名 Z に対し, X is-a Y ならば, Z of X is-a Z of Y である。

推移律: 任意の属性式 X, Y, Z に対し, X is-a Y かつ Y is-a Z ならば X is-a Z である。

例えば、「ichiroh is_a human」ならば「hand of ichiroh is_a hand of human」であり、さらに「human is_a mammal」ならば「ichiroh is_a mammal」である。

このような is-a 関係の推論規則は、DOT[8], Quixote[10], ITL[2] などでも使われており、一般的なものであると考えられている。

is-a 関係式間のルールを記述することもできる。例えば、「人間であり男でないものは女である」というルールは次のように記述する。

```
if X is_a human, X is_not_a male  
then X is_a female.
```

ドメイン集合、アドレス、RDI、隣接ドメインについての知識は、特定のオブジェクト名や属性名を用いて次のように記述する必要がある。

```
member of company_A is_a  
[section_1, section_2].  
rd_A is_a member of section_1.  
rdi of rd_A is_a 'XX:YY:102a'.  
address of rd_A is_a  
['XX:YY:10:*','XX:YY:20:*'].  
adjacent_domain is_a  
[rd_C, rd_D, rd_F].
```

ドメイン集合のメンバとして他のドメイン集合を指定することができる。

ドメインのアドレスとは、そのドメインに属すすべてのホスト (エンドシステムと中継システム) のアドレスプレフィックスである。ドメイン集合についても、適切なアドレス・プレフィックスがある場合、これを明示することで経路制御の効率を上げることができる。アドレスの定義は、上記のように略式メンバ定義の構文で行う必要がある。

3.2 経路制御ポリシーの構文

経路制御ポリシーはその内容によって経路選択ポリシー、経路集約ポリシー、中継ポリシーに分けられる。

3.2.1 経路選択ポリシー

パケットを配送する際に特定のドメインを通る経路を避けたいというポリシーは、次の構文で記述する。

```
avoid_route_through( 通過ドメイン 1)  
[rather_than_through( 通過ドメイン 2)]  
[destined_to( 目的地)].
```

角括弧 ([]) で囲まれた部分は省略可能である。

同様に、特定のドメインを通る経路を優先したいというポリシーは、次の構文で記述する。

```
prefer_route_through( 通過ドメイン 1)  
[rather_than_through( 通過ドメイン 2)]  
[destined_to( 目的地)].
```

(通過ドメイン 1), (通過ドメイン 2), (目的地) には、ドメイン名、ドメイン集合名、あるいはそれらのリストを記述できる。また、(通過ドメイン 1), (通過ドメイン 2) には、「ドメイン名-ドメイン名」という構文でドメイン間の回線を指定することもできる。

例えば、「rd_1へパケットを送る際にrd_2を通る経路よりrd_3またはrd_4を通る経路を優先したい」というポリシーは、

```
prefer_route_through [rd_3, rd_4]  
rather_than_through rd_2  
destined_to rd_1.
```

と記述する。

優先して通るべきドメインと避けるべきドメインの両方を通る経路に対しては、避けるほうのポリシーが優先される。また、目的地としてより少数のドメインを指定したポリシーが優先される。

経路制御の自動設定ツールでは、これらのポリシーを踏まえて、より安全で、より速く伝送できる、よりコストの安い経路が選択されるように設定がなされる。

3.2.2 経路集約ポリシー

経路情報を隣接ドメインに配布する際に、ある目的地への経路情報を集約して配布したいというポリシーは、次の構文で記述する。

```
aggregate_route destined_to(目的地)
[for(隣接ドメイン)].
```

(目的地)にはドメイン集合名を指定する。(隣接ドメイン)は、集約した経路情報を配布する隣接ドメインのドメイン名、あるいはそのリストである。省略した場合すべての隣接ドメインに配布する。

二つの経路集約ポリシーがあって、ポリシーAの(目的地)のアドレス空間がポリシーBのそれに含まれている場合、ポリシーAの対象となる経路はポリシーBの対象となる他の経路とは別に集約され、配布される。例えば、

```
address of rd_A is_a ['XX:YY:10:*'].
address of company_A is_a ['XX:YY:*'].
```

```
aggregate_route_to rd_A for rd_D.
aggregate_route_to company_A.
```

と記述した場合、隣接ドメイン rd_D へ経路情報を配布する際に、目的地のアドレス・プレフィックスが XX:YY:10:* である経路情報が集約され、それ以外の経路情報のうち目的地のプレフィックスが XX:YY:* であるものも別に集約される。また、rd_D 以外の隣接ドメインへ経路情報を配布する際には、目的地のプレフィックスが XX:YY:* である経路情報が集約される。

3.2.3 中継ポリシー

どのようなパケットを中継するか、あるいは中継しないかというポリシーは、次の構文で記述する。

```
prohibit_transit_packet
[from(発信元)][to(受信先)].
allow_transit_packet
[from(発信元)][to(受信先)].
```

prohibit_transit_packet 文は(発信元)から(受信先)へのパケットを中継しないというポリシーを表す。allow_transit_packet 文は反対に、(発信元)から(受信先)へのパケットを中継するというポリシーを表す。いずれの場合も、発信元、または受信先を省略でき、省略した場合任意と解釈される。発信元と受信先の両方を省略した場合、他のいずれの中継ポリシーにも当てはまらないパケットの扱いを指示することになる。デフォルトでは、このようなパケットに対しては中継が行われる。

矛盾するポリシーがあった場合、(受信先)のドメイン数が少ないほうが優先される。(受信先)のドメイン数が同じ場合、(発信元)のドメイン数が少ないほうが優先される。例えば、

```
prohibit_transit_packet to [rd1,rd2,rd3].
allow_transit_packet from [rd4] to rd2.
```

と書いた場合、任意のドメインから rd1 と rd3 へのパケットと、rd4 以外から rd2 へのパケットは中継しない。

3.2.4 ルール形式のポリシー

ルールの結果部として上記のポリシーを記述することができる。例えば、「ドメイン A へのパケットを配送する際に質の低い回線は避けたい」というポリシーは

```
if X is_a low_quality_circuit
then avoid_route_through X
destined_to domain_A.
```

というように記述する。これに対し、知識ベースに

```
domain_B-domain_C
is_a low_quality_circuit.
```

という関係式が格納されていた場合、このルールから

```
avoid_route_through domain_B-domain_C
destined_to domain_A.
```

という具体的なポリシーが導出される。

3.3 その他

“%” から行末まで、および “/*” から “*/” までではコメントと解釈され、無視される。

複数の BIS やドメインで共通の知識やポリシーを用いる場合、これらを別のファイルにまとめ、次の構文で読み込むことができる。

```
include '〈ファイル名〉'.
```

4 経路制御自動設定ツールの実装

ポリシー記述言語によって記述された知識とポリシーから IDRP のポリシー設定ファイルを自動作成するツールを、SWI-prolog Ver. 1.8 [9] を用いて Sun* ワークステーション上に実装した。

図 1 に、このツールにおける処理の流れを示す。

ユーザの記述したポリシーと知識ベースは prolog の節として読み込まれる。is-a 関係推論機構によって知識ベースとルール形式のポリシーから具体的なポリシーが導出され、非ルール形式のポリシーとともに設定ファイル作成機構によって IDRP の設定に変換される。IDRP の設定を作成する際には、アドレスや RDI、隣接ドメインやドメイン集合などについての知識が用いられる。

is-a 関係推論機構は prolog インタープリタの問い合わせ処理機構をそのまま利用している。継承と推移律

*Sun は米国 Sun Microsystems 社の商標である。

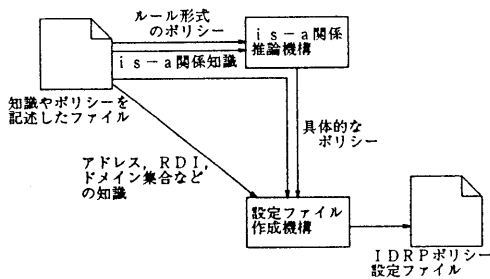


図 1: 経路制御自動設定ツールの処理の流れ

はあらかじめルールとして定義されており、これにユーザ定義のルールを加えた状態で、経路選択、経路集約、中継に関する各ポリシーについて問い合わせを行い、導出可能なすべての具体的なポリシーが求められる。

```

% 共通知識のインクルード
include 'common_knowledge_file'

% 隣接ドメインの定義
adjacent_domain is_a [rd_C,rd_D,rd_F].

%%%% 経路選択ポリシー %%%
prefer_route_through section_B
  destined_to section_C.

%%%% 経路集約ポリシー %%%
if X is_a adjacent_domain,
  X is_not_a our_domain
  then aggregate_route
    destined_to company_C for X.

%%%% 中継ポリシー %%%
if X is_a competitor
  then prohibit_transit_packet from X.
if X is_a competitor
  then prohibit_transit_packet to X.

```

リスト 2: 知識とポリシーの記述例

5 考察

5.1 ポリシー記述言語について

今回実装したポリシー記述言語によってリスト 1 の設定を記述したものを、リスト 2 に示す。アドレスや RDI、ドメイン集合などの知識は組織内の複数のドメインで共有できるので、別のファイルにまとめ、インクルードしている。このインクルードファイルの内容をリスト 3 に示す。

%%%% 知識ベース %%%

% 組織の階層構造の定義

```

member of company_A is_a
  [sectionA_coA, sectionB_coA].
  member of sectionA_coA is_a [rd_A].
  member of sectionB_coA is_a [rd_B].
member of company_B is_a [rd_D].
member of company_C is_a
  [section_A, section_B, section_C].
  member of section_A is_a [rd_E, rd_F].
  member of section_B is_a [rd_C].
  member of section_C is_a [rd_G].

```

% RDI(RDCI) の定義

```

rdi of rd_A is_a 'XX:YY:102a'.
rdi of rd_B is_a 'XX:YY:000b'.
rdi of rd_C is_a 'XX:WW:2201'.
rdi of rd_D is_a 'XX:ZZ:1024'.
rdi of rd_E is_a 'XX:WW:5100'.
rdi of rd_F is_a 'XX:WW:5b01'.
rdi of rd_G is_a 'XX:WW:000i'.

```

% アドレスの定義

```

address of rd_A is_a ['XX:YY:10:*'].
address of rd_B is_a ['XX:YY:0:*'].
address of rd_C is_a ['XX:WW:2:*'].
address of rd_D is_a ['XX:ZZ:*'].
address of rd_E is_a ['XX:WW:51:*'].
address of rd_F is_a ['XX:WW:5b:*'].
address of rd_G is_a ['XX:WW:0:*'].
address of section_A is_a ['XX:WW:5:*'].
address of company_A is_a ['XX:YY:*'].
address of company_C is_a ['XX:WW:*'].

```

% その他の知識

```

company_A is_a competitor.
if X is_a member of company_C
  then X is_a our_domain.
if X is_a member of Y,
  Y is_a our_domain
  then X is_a our_domain.

```

リスト 3: 共通知識の記述例

リスト 1 の設定はアドレスや RDI などの数値でポリシーの対象を記述していたため、可読性が低かった。また、ポリシーの記述の順序が優先順位を表すため、新たなポリシーの追加が難しかった。

このポリシー記述言語では、組織の構造を反映した階層構造のドメイン集合を定義して、このドメイン集合に対してポリシーを記述することで、可読性を向上している。

また、ポリシーの優先順位がシステムによって決定されるので、例外やデフォルトとなるポリシーを容易に記述・追加することができる。たとえば、A 社はライバル企業なのでルール形式のポリシーから「A 社へのパケットは中継しない」というポリシーが導出されるが、これに対する例外として「当社 (C 社) の A 部門から A 社の A 部門へのパケットは中継したい」というポリ

シーを記述したい場合、

```
allow_transit_packet
  from section_A to sectionA_coA.
```

と書き加えるだけでよい。

ルール形式によるポリシーの記述は、知識とポリシーの独立性を高め、保守を容易にすることができる。例えば、リスト2では「社内への経路の情報を社外の隣接ドメインに配布する際には経路情報の集約を行う」というポリシーを、

```
if X is_a adjacent_domain,
  X is_not_a our_domain
then aggregate_route
  destined_to company_C for X.
```

と記述しているため、ネットワークポロジが変化しても、対応する知識を更新するだけで適切な再設定を行うことができる。

このポリシー記述言語では、ポリシーの対象の広さ(ドメイン数)によって優先順位を付ける。しかし、対象の広さによらず優先的に実現したいポリシーがあった場合、ルール形式のポリシーによって導出された具体的ポリシーやインクルードファイルに記述されたポリシーなどによってこのポリシーが部分的に反映されない恐れがある。このため、今後オプションとしてポリシーの優先度を明示できるようにする必要がある。

5.2 is-a 関係推論機構について

今回実装したツールでは、is-a 関係推論機構として prolog インタープリタの問い合わせ処理機構をそのまま用いた。このため停止性が保証されず、また、知識ベースが大きくなるにつれて処理が極端に遅くなるという問題がある。

例えば前節の例に、

```
if X is_a competitor, Y is_a friend of X
then Y is_a competitor.
```

というルールを加えると、「X is_a competitor」という問い合わせが無限に繰り返され、推論が停止しなくなる。

is-a 関係によって表現された知識に対して、is-a 関係の順序性と継承関係に着目して推論を行う枠組み DOT [7] が提案されている。DOT における推論アルゴリズムは探索によるものではなくオートマトンを用いたものであり、問い合わせに対して正規表現を用いて答える。このため答が無限集合になる問い合わせに対しても答えることが可能であり、推論の停止性が保証されている。また、この推論アルゴリズムの高速化の手法も提案されている [6]。そこで、今後は is-a 関係の推論に DOT を用いることが有効であると考えられる。

6 おわりに

本研究では、経路制御に対する抽象的なポリシーと設定に必要な知識ベースを記述するための言語を設計した。また、この言語で記述されたポリシーから IDRP の設定を行うツールを実装することによって、抽象的なポ

リシーからの経路制御の自動設定を実現した。その結果、経路制御の設定・保守作業を簡素化することができた。また、設定作業におけるケアレスミスを防ぎ、設定の信頼性を向上することができた。

今回の実現では、経路制御における必要最小限の機能のみに限定したが、今後、国際標準プロファイル (ISP) の開発が行われたり、大規模ネットワークにおける運用経験を積むなど、IDRP の実装や運営の範囲が明確になると、その仕様や要求条件に応じて言語の表現力と自動設定ツールの機能を強化し、より複雑なポリシーを扱えるようにする必要がある。

参考文献

- [1] Estrin, D.: "Policy Requirements for Inter-Administrative Domain Routing," *Computer Networks and ISDN Systems*, No. 22, pp. 179-191 (1991).
- [2] N. Guarino: "A Concise Presentation of ITL," *ACM-SIGART BULLETIN*, Vol. 2, No. 3, pp. 61-69 (1991).
- [3] Lougheed, K. and Rekhter, Y.: "Border Gateway Protocol(BGP)," RFC1163 (June 1990).
- [4] 村山優子, 曾根文樹: "政策的経路制御とその動向," 情報処理学会マルチメディア通信と分散処理研究会報告書, Vol. 93, No. 36, pp. 1-7 (May 1993).
- [5] Rekhter, Y.: "Inter-Domain Routing Protocol(IDRP)," *Internetworking: Research and Experience*, Vol. 4 (1993).
- [6] 清 一隆, 塚本昌彦, 西尾章治郎: "ドット記法と IS-A 関係を用いた推論システムの高速度に関する研究," 第 8 回人工知能学会全国大会論文集, pp.517-520 (1994).
- [7] Tsukamoto, M., Nishio, S. and Fujio, M.: "DOT: A term representation using DOT algebra for knowledge-bases," in *Proc. of the 2nd Int'l Conf. on Deductive and Object-Oriented Databases*, pp. 391-410 (1991).
- [8] Wielemaker, J.: "SWI-Prolog 1.8 Reference Manual" (1993).
- [9] H. Yasukawa, H. Tsuda, and K. Yokota: "Objects, Properties, and Modules in Quixote," in *Proc. of the Int'l Conf. on Fifth Generation Computer Systems*, pp.257-268 (1992).
- [10] ISO/IEC/JTC1/SC6 DIS 10747 "Protocol for Exchange of Inter-domain Routing Information among Intermediate Systems to Support Forwarding of ISO 8473 PDUs" (1993).