情報ネットワークシステムのポリシー制御 “PolicyComputing” の適用と実装

菅野政孝[†] 田中俊介[†] 坂田祐司[†]
小熊慶一郎[†] 白鳥則郎[††]

インターネットのビジネスでの利用が多様化すると，たとえば通信制御やセキュリティの設定レベルなどのリソース配分についてもユーザごとに様々な利用方法が要求されるようになる．このとき，ユーザの多様な要求に対してリソースの配分をどのような基準・考え方によって行うかというルールがポリシーであり，このポリシーに従ってユーザに有効にリソースを配分する方法がポリシー制御である．従来ポリシー制御ではポリシーをディレクトリサーバに格納し一元管理することが一般的であったが，ポリシーの数が多い場合にはパラメータの設定など管理コストがきわめて大きいことが問題であった．本論文ではリソースを管理するためのポリシー（個別ポリシー）の上位概念として管理者にとって設定しやすい「マスターポリシー」を定義し，これを用いて個別ポリシーを自動生成することによって管理コストを削減する仕組み「PolicyComputing」を考察した．また，これを通信品質制御やセキュリティ管理のアクセス制御へ応用することにより有効性を評価した．さらに PolicyComputing を実装し動作させることによりシステム管理者の管理コストを削減できることを確認した．

Application and Implementation of Policy Control Method “PolicyComputing” in Computer Networks

Masataka Sugano,[†] Shunsuke Tanaka,[†] Yuji Sakata,[†]
Keiichiro Oguma[†] and Norio Shiratori[††]

The more usage of Internet in the business area becomes various, the more resource allocation such as a communication bandwidth or a security level is required to be various according to user levels. The resource allocation will be done under rules of some criteria or thoughts, which are called “Policy”, while effective methods of the resource allocation to every users are called “Policy Control”. Formerly policies were generally stored in the “directly servers” where they could be managed in a same way in a system. In this case, costs of setting up parameters for systems were extremely large and this has been one of the biggest problems for the system installation. In this paper, we define master policies, which are easy to set up for the system by installers, and discuss about the method “PolicyComputing” for eliminating the cost of installing the system, as well.

1. はじめに

近年，インターネットがインフラストラクチャとして重要な位置を占めてくるに従い，企業情報システムやエレクトロニックコマースに利用される事例が急速に増加・多様化してきている．

ビジネス利用が多様化するとたとえば通信帯域やセキュリティの設定レベルなどもユーザごとに様々な利用方法が要求される．しかしながら有限なリソースの範囲内ではアプリケーションプログラム（以下，AP と略す）間でリソース確保が競合することは避けられない．このとき，インターネットのようなベストエフォート型のネットワークにはこれらの競合を制御するメカニズムが備わっていないため，上位のプログラムでこのような制御を実現する必要がある[1)]．

AP の緊急度により実行優先度をつけそれに対応して通信帯域を割り当てたり，ユーザによってアクセスできるリソースに差をつけるセキュリティレベルの制御を行ったりするためには複数の AP をネットワーク内で観察し AP 間の優先度やリソースの配分をコントロールする仕組みが必要である．また，それと同時に

† 株式会社 NTT データ
NTT DATA Corporation
†† 東北大学電気通信研究所
Research Institute of Electrial Communication, Tohoku University

運用時にこのような各 AP の実行優先度やリソース配分をどのような基準・考え方で行うかというルールを決めておく必要がある．

ここであらかじめ決めておくルールがポリシーであり，こういったメカニズムで IP ネットワークのリソースを有効に活用しユーザの多様なニーズにこたえるのがポリシー制御である[2]．

通信品質のポリシー制御についてはインターネットに関する標準化グループである IETF（Internet Engineering Task Force）の RAP（Resource Allocation Protocol）WG において検討がなされている．

ここでは，「イベントに対する処理の判断（ポリシー）をサーバへアウトソーシング（クライアントから問合せを出して応答を受ける）するための仕組み」やそのときに必要となるプロトコルを検討しており，関連 RFC（2748〜2753）が発行されている．

さらに，IETF ではこれらのポリシーをすべてディレクトリサーバに格納することで，きめ細かいシステム管理を実現するための概念（DEN: Directory Enabled Network）に基づきディレクトリのスキーマ（どの情報をディレクトリサーバ内のどこに格納するか）に関する標準仕様を作成してきた[3)〜5)]．

DEN を利用するとすべてのポリシーおよびリソースの設定情報をディレクトリ上で一元管理できるようになる．これにより以下のような点から「各リソースの設定に必要な管理コスト」を軽減することができる．

- 同一の情報を複数回入力する必要がなくなる．
- すべての情報に同一の手段でアクセスできる．

しかし，DEN を利用するとしても，設定する項目の内容自体は変わらないので，以下の 2 つの問題点が残っている．

- 設定する項目の数が多い．
- システムの知識が豊富な人でなければ管理できない．

すなわち，DEN を利用した場合でも「各リソースの設定に必要な管理コスト」は多く残っているといえる．

管理コストを減らす方法としてはポリシーの策定自体を容易とする方法が考えられ，セキュリティポリシーの策定をツールで実現した例もある[6]．しかし，この場合でもシステム知識の豊富な人が管理する必要性がある．

筆者らは「各リソースに必要な管理コスト」を削減することを狙いとして，一般の管理者でも容易に設定できる表現形式であるマスターポリシーによって，情報システム内の機器，サーバプロセス，ユーザなどすべてのリソースを一元的に管理するための仕組み「PolicyComputing」と実際のネットワークへの応用方法を検討してきた[7),8)]．

本論文ではポリシー制御を実現するための PolicyComputing の考え方とその応用方法を提案する．また，PolicyComputing を具体的にセキュリティのアクセス制御に応用し実装・評価を行った．

以下 2 章でポリシー運用管理コスト削減のための仕組みである PolicyComputing についてその考え方と機能を提案する．3 章で PolicyComputing を通信品質（QoS: Quality of Service）制御，およびセキュリティ管理へ適用した例を，4 章で PolicyComputing を適用したセキュリティのアクセス制御について実装結果と評価を示す．5 章でまとめと今後の課題を示す．

2. PolicyComputing の提案

2.1 目　　的

PolicyComputing の目的は，運用管理コストを削減し，少ない運用管理コストでもきめ細かいシステム管理（資源の効果的な活用，セキュリティの向上など）ができるようにすることである．

各リソースの設定に必要な管理コストを削減する考え方は以下のとおりである．

- マスターポリシーから個別ポリシーを自動的に作成する．これにより管理者は少数のマスターポリシーを入力すればよくなる．
- マスターポリシーは管理者にとって理解しやすい表現形式で記述できる．これによりシステムに精通していない管理者でも十分に管理が行えるようになる．

2.2 PolicyComputing の機能

PolicyComputing を適用するネットワークシステムのイメージ図を**図 1** に示す．

PolicyComputing でマスターポリシーに基づく一元管理を実現する機能は**図 2** のとおりである．

図 2 の各要素の役割を以下に示す．

- 管理者
  PolicyComputing が対象とするネットワークシステムの管理者
- Governor
  管理者とディレクトリサーバ，サービスコントローラとのインタフェース
- Directory サーバ
  マスターポリシー，個別ポリシー，リソースのプロファイル，知識データを格納しておくデータベースである．PolicyComputing ではこれらの情報を一元管理できるよう LDAP を利用する．

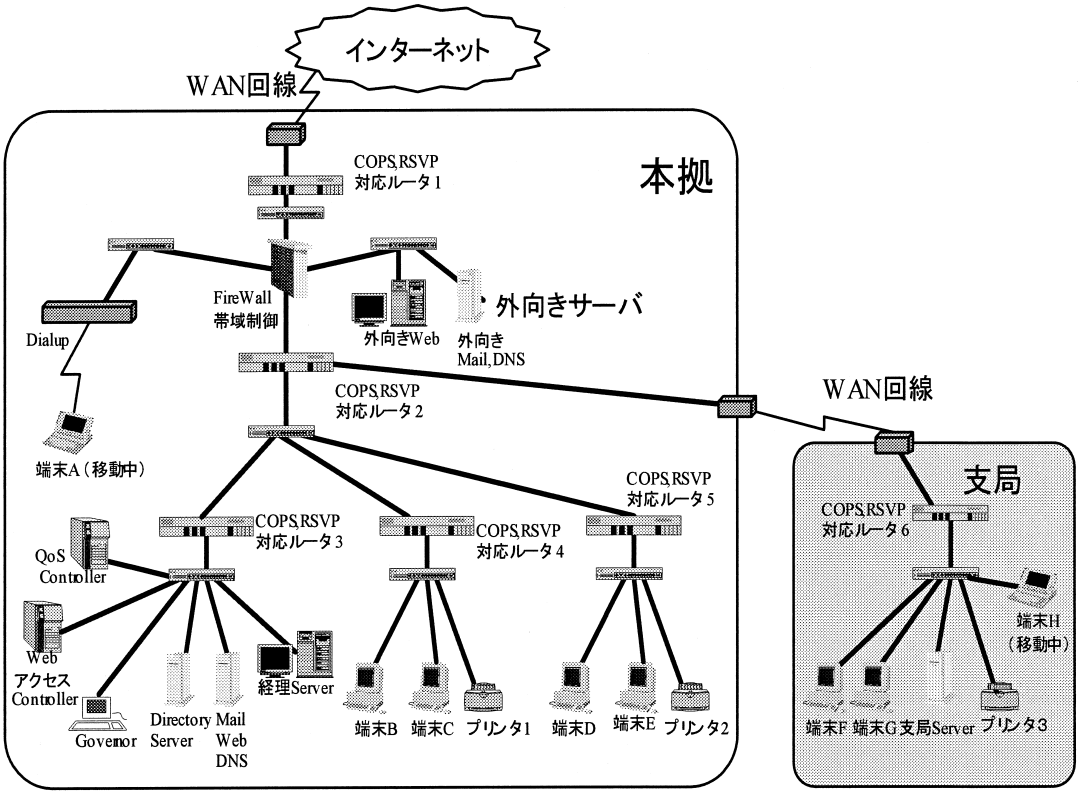

図1 ネットワークシステムイメージ図
Fig. 1 Network system model.

- サービスコントローラ
  QoS コントローラ，Web アクセスコントローラなどサービス種別ごとにそれぞれのサービスコントローラがある．
- 管理リソース
  PolicyComputing が対象とするリソースである．

また，図 2 の ①〜⑨ までのそれぞれの処理内容を以下に示す．

① 管理者がリソースに「PolicyComputing に加えるための最小限の設定」を行う．

リソースを PolicyComputing に組み込む場合には，リソースがディレクトリサーバと通信できなければならない．そこで，管理者はリソースのアドレス，ディレクトリサーバ（もしくはポリシーサーバ）のアドレスなどをリソースに設定する．

② 各リソースがプロファイルを自動的に格納する．

各々のリソースは起動時に自分が所持している動作状況をプロファイルとしてディレクトリサーバに格納する．また，起動中に動作状況に変化が生じた場合には変化分の情報をディレクトリサーバに格納する．プロファイルとは以下のようなデータである．

［プロファイルの例］

- 属性：ファイルの情報種別
  内容：ファイル X は人事情報
- 属性：端末にログインしているユーザ
  内容：Host A にユーザ Y がログインしている．

リソースの中には，ディレクトリサーバと直接通信できないものがある．ディレクトリサーバと直接通信できないリソースは，ポリシーサーバなどを介してプロファイルをディレクトリサーバに格納する．

③ 管理者がプロファイルを格納する．

プロファイルの中には，管理リソースが自動的（機械的）に格納することができないプロファイルがある．たとえば，ユーザ・アカウントなどである．自動的（機械的）に格納できないプロファイルは，管理者が手作業でディレクトリサーバに格納する．

④ 知識データが登録されている．

知識データとはシステムに依存しない運用ノウハウのような情報である．システム管理者はあらかじめディレクトリサーバに知識データを格納しておき，システム運用中には変更する必要がないようにしておく．知識データとは以下のようなデータである．

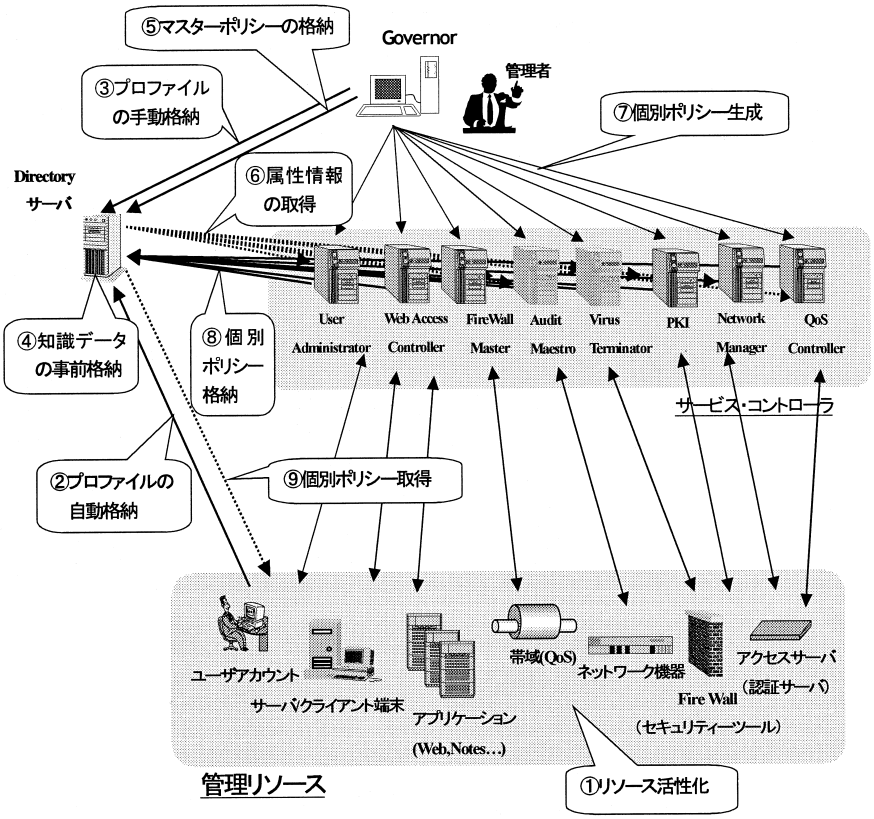

図 2 PolicyComputing の処理内容
Fig. 2 Processing method of PolicyComputing.

［知識データの例］

- 属性：アプリケーションのポート番号
  内容：ポート番号 N 番は IP-Phone である．
- 属性：サービスの品質
  内容：IP-Phone は 100 kbps の帯域があれば高品質である．

⑤ 管理者がマスターポリシーを格納する．

マスターポリシーも「属性」と「内容」によって構成されている．内容は「A = B」という形式に容易に変更できるような文章である．属性の値によって A および B に入る単語の種類を定義しておく（たとえば，「属性」が「文書セキュリティ」である場合は A には情報別とユーザグループの組が入り，B には閲覧の可否が入る）．マスターポリシーとは以下のようなデータである．

［マスターポリシーの例］

- 属性：文書セキュリティ
  内容：人事情報は管理職のみ閲覧可とする．
- 属性：サービス QoS ポリシー
  内容：管理職のコミュニケーション・ツールの通信は高品質にする．

⑥ マスターポリシー，プロファイル，知識データのうち必要な情報を取得する．

ディレクトリサーバ内のどの属性の情報を参照するかは，サービス・コントローラごとに異なっている．サービス・コントローラにはそれぞれ必要なマスターポリシー，プロファイル，あるいは知識データなどの属性情報が定義されていて，管理者によりあらかじめ設定されておくものとする．サービス・コントローラは自分にとって，必要な属性のデータのみを取得する．

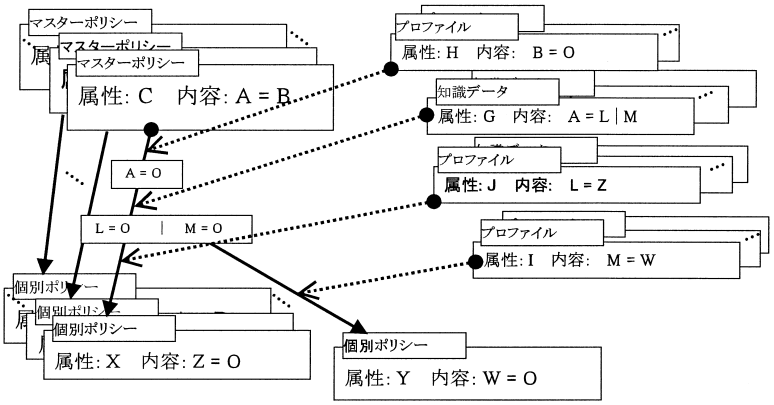

図 3 個別ポリシーの生成方法
Fig. 3 Method of generating the individual policy.

⑦ 取得した情報を利用して個別ポリシーを自動生成する．

個別ポリシーも「属性」と「内容」によって構成される．内容はリソースの詳細な設定情報である．個別ポリシーとは以下のようなデータである．

［個別ポリシーの例］

- 属性：ファイル・アクセス
  内容：ユーザ Y は xxx にアクセス不可
- 属性：ネットワーク QoS
  内容：端末 A–端末 B 間のポート N 番の通信を 100 kbps で帯域を確保する．

個別ポリシーの作成は，数式 A に定理 X を適用すると数式 A と等価で表現が異なる数式 B が作成されるという「数式の変換」に似ている．マスターポリシーにプロファイル・知識データを適用して，マスターポリシーと等価で表現が異なる個別ポリシーを作成する．数式の変換では複数の数式を変換させて 1 つの数式を作成する場合が多いが，ポリシーの変換（個別ポリシーの作成）では，少数のマスターポリシーを変換させて多数の個別ポリシーを作成する場合が多くなる（**図 3** 参照）．

プロファイルの中には，「端末を利用しているユーザ名」など，現在の動作状況を表すものがある．こうしたプロファイルの内容が変化した場合には，ポリシー変換をやり直して，関係がある個別ポリシーを作成し直す．

どのマスターポリシーからどのプロファイル・知識データをどのように利用して個別ポリシーを生成するかは，サービス・コントローラごとに異なっている．本論文では，QoS Controller における個別ポリシーの生成方法についてのみ示す（**図 4** 参照）．

⑧ 個別ポリシー（設定情報）をディレクトリサーバに格納する．

⑦ で作成した個別ポリシーをディレクトリサーバの所定の場所に格納する．

⑨ 各リソースが個別ポリシー（設定情報）を取得する．

リソースは自分にとって，どの属性の個別ポリシーが必要であるかを知っている（あらかじめ定義されていて，前もってリソースに設定しておく）．ディレクトリサーバ内のどの属性の情報を参照するかは，リソースごとに異なる．たとえばある属性の情報はルータと Firewall という 2 つのサービス・コントローラから参照されるが，ある属性の情報は端末からしか参照されない場合もあり，同種類のリソースでも異なる属性の情報を参照する場合もある，などである．

3. PolicyComputing の応用例

3.1 通信品質制御への応用

（1）通信品質の管理におけるポリシー制御の導入

組織内には様々な業務が存在し，これらの業務が扱う情報もまた，数値データ，テキスト，画像や映像など多岐にわたる．これらの情報がネットワーク内で転送されるとき，業務の重要性や緊急度合い，情報の大きさなどによって通信 AP に異なった転送速度や通信帯域を付与することによりネットワークを有効に活用することができる．このためネットワークのトラフィックの状況によって各通信 AP にどの程度の通信品質を与えるかという通信品質制御の考え方（通信品質制御ポリシー）を規定しておき，これに基づきシステムの管理を行うことが一般的となってきている．

（2）通信品質制御の一元管理の有効性

ネットワーク内で複数の通信 AP にどの程度通信品

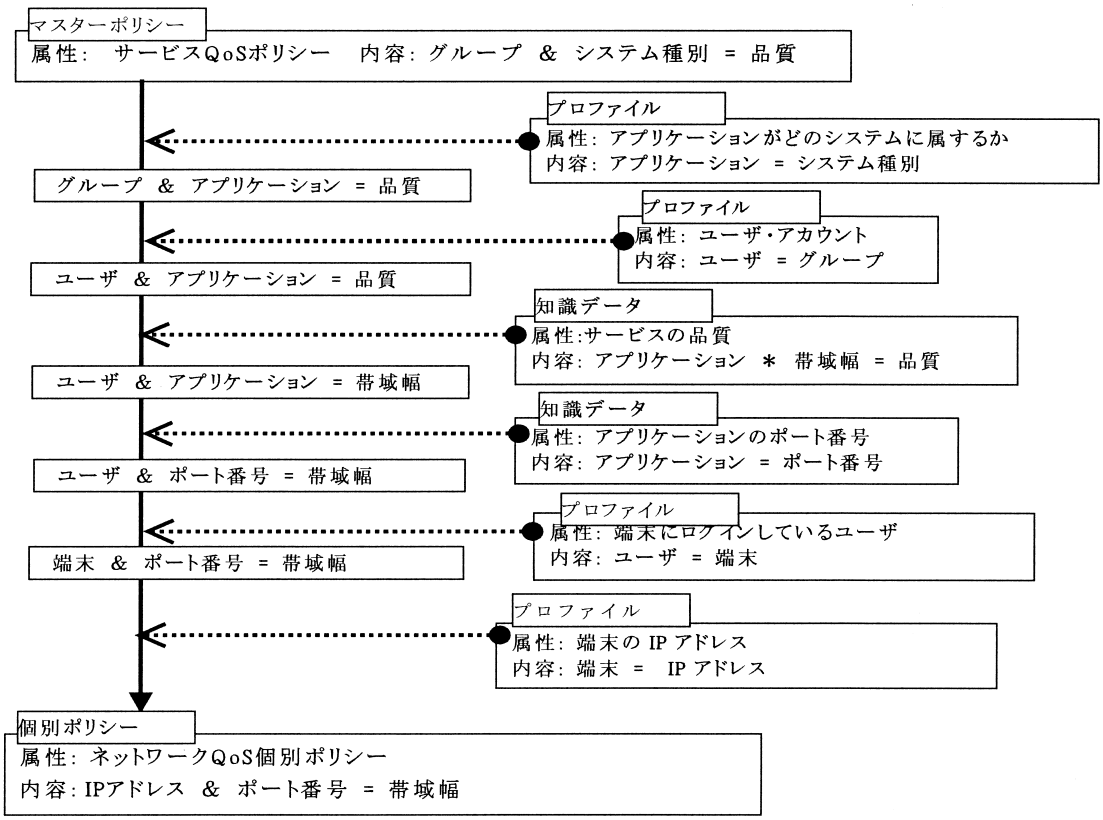

図 4 QoS 個別ポリシーの生成方法
Fig. 4 Method of the QoS individual policy.

質を与えるかということを決めるためには，たとえばネットワーク内での通信 AP のトラフィック状況を監視し，ネットワークの状況に合わせて各通信 AP にどの程度の通信品質を与えるかというポリシーを決めておく必要がある．また通信 AP がネットワーク内で相互に矛盾なく動作するためには，このときに適用するポリシーが同一の考え方に基づくものでなくてはならない．このため，ネットワーク内で適用する通信品質制御ポリシーは一元的に管理する必要がある．

これにより，通信品質制御ポリシーに PolicyComputing を適用することは非常に有効である．

(3) PolicyComputing の通信品質制御ポリシーへの適用

PolicyComputing を応用する場合の例として，通信品質制御ポリシーを管理する場合を図を用いて説明する（**図 5** 参照）．

この例では管理職社員がある業務（0A）を行うとき，情報は高品質で通信するものとする．

この場合システム構築時には以下の処理を行う．

(1) 管理者はマスターポリシーを Governor に入力する．

(2) QoS Controller がマスターポリシーから通信品質個別ポリシーを生成する（図 4 参照）．

(3) ポリシーサーバは通信品質個別ポリシーをディ

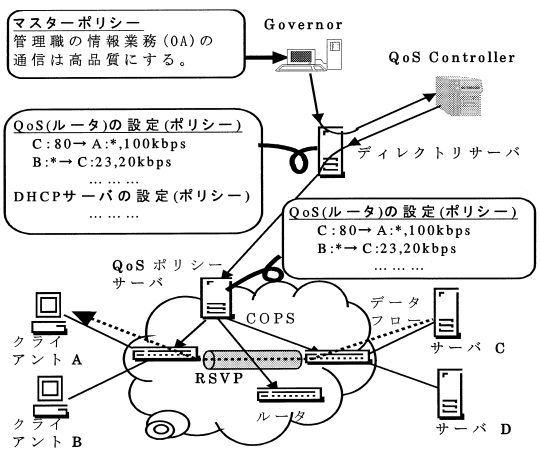

図 5 PolicyComputing の通信品質制御への応用
Fig. 5 Application of PolicyComputing to the communication quality control.

表 1 アクセス制御に関するマスターポリシーの例
Table 1 Examples of master policy for access control.

情報種別項目 ＜属性＞		セキュリティポリシー ＜内容＞	
大項目	小項目	機密レベル	開示範囲
研究開発成果	研究成果	厳秘	グループ内
	研究テーマ	秘密	研究所内
	実験結果	秘密	研究所内
	報告書	社外秘	社員全員

レクトリサーバから読み出す．

仮に，PolicyComputing を利用しないとすれば，管理者は，図 4 のフローに相当する作業を自分で行い，直接ディレクトリサーバに通信品質制御のポリシーを設定する必要があり，負担が大幅に増大する．

3.2 セキュリティ管理への応用

(1) セキュリティ管理におけるポリシー制御の導入

業務の効率化のために組織内に分散した各種情報を相互に参照することが必須になってきている．特にイントラネットのように基本的にクライアントサーバシステムで構成されたネットワークでは組織内の機密情報をはじめとする重要な情報が各サーバ間に分散しているため情報の適切なアクセスの必要性が認識されている．

具体的なアクセス制御の例としては以下のようなものがあげられる．

- WWW やメールサーバ（社内，社外向け）へのアクセスの制御
- 各サブ組織が持っているサーバ類のアクセス制御（サブ組織内，あるいは他サブ組織間，組織外）
- 各自が持つファイルのアクセス制御，等々

このようなアクセス制御が必要なシステムではセキュリティの管理を厳格に行うことが急務である．

セキュリティ管理に関しては企業などそれぞれの組織が管理における考え方（セキュリティポリシー）を規定し，これに基づきシステムの管理を行うことが一般的となっている．

社内で利用する情報に関するアクセス制御に関するマスターポリシーの例を**表 1** に示す．

分散環境下でセキュリティポリシーを適用するには以下のような仕組みが必要であると考えられる．

- セキュリティポリシーを集中的に管理することが可能な仕組み
- 分散された環境にそのセキュリティポリシーを反映することが可能となる仕組み
- セキュリティ管理者が自分の決めたポリシーが反映されていることが把握できる仕組み

上述の仕組みを導入する場合，最も問題となるのがコストの増大をいかに回避するかということである．コスト増大の要因を以下に示す．

- リソースが分散されている場合，これらのリソースが格納されているアドレスをすべて知っている必要がある．
- 各サブ組織のシステム管理者が別々にセキュリティ制御に関するパラメータの設定を行わなければならない．
- また，管理者がリソースの相違を知っている必要がある．
- 新しいリソースを追加するたびに各管理者が新規にセキュリティパラメータを理解しシステムに追加する必要がある．

たとえば，Web サーバと Firewall で同一アクセス制御を行いたいときは，セキュリティパラメータを Web サーバ用から Firewall 用に変換する必要があるなどが考えられる．

(2) セキュリティ情報の一元管理の有効性

（1）で示した問題を解決するためにはこれらセキュリティ情報を一元管理することが有効である．

以下にセキュリティポリシーおよびユーザ情報の一元管理による有効性を示す．

（A）セキュリティポリシーの一元管理の有効性

ネットワークセキュリティの対象となる脅威には盗聴，改竄，なりすまし，不正アクセスなどがあげられるが，対策としては暗号化などのように情報自体に対策を施すものと，ネットワークを通してリソースに不正にアクセスされることを防ぐための対策を施すものとある．今回，ネットワークの管理の観点からアクセス制御に着目した場合の問題は以下のとおりである．

たとえば WWW サーバを例にとった場合，ユーザの異動やセキュリティポリシーの変更，情報の追加などに対してセキュリティ管理者から WWW サーバ管理者に対し変更の依頼を行うが，依頼者と作業者が異

なるため確認に稼働がとられたり，相互の行き違いからトラブルが発生したりすることが少なくない．

これらは分散システムでサーバが各所に設置されている場合にはなお顕著となる．

したがって，これらの問題を解決するには，i) 分散されたサーバ，クライアントなどのリソースに対して，集中的な管理を可能とする，ii) 管理者がリソースの相違，物理的・論理的な位置を意識することなくパラメータを設定することを可能とする，iii) 新しいリソースが追加されても管理上の変更を容易とする，ことが望ましい．

このためにはサーバなどのリソースの情報を一元的に管理する仕組みを作ることが有効となる．

（B）ユーザ情報の一元管理の有効性

ネットワークシステムの中では，実際のユーザが持つ氏名がたとえばディレクトリサーバ，ドメインのアカウント，IP アドレスなどで異なって（違った形式で）管理されている．このため，システム運用者，セキュリティ管理者，そしてときには本人がこれらの情報を間違いなく管理するにはかなりの労力を要する．また，組織では人事異動などが頻繁に発生するがこの場合の管理稼働はきわめて莫大なものとなる．

したがって，これらの問題を解決するには管理元が異なる場合でも同じユーザである限りは一元的に管理する仕組みを作ることが有効となる．

（3） PolicyComputing のセキュリティポリシーへの適用

PolicyComputing の特長はポリシーを一元化すること，および管理の対象となる情報を一元管理することである．

以上から（2）で示したとおりセキュリティポリシーに PolicyComputing を適用することはきわめて有効であると考えられる．

この場合，i) ポリシーをどのように一元化するか，ii) 管理の対象となるセキュリティポリシーとユーザ情報をどのように一元管理するか，ということが課題となる．

以下にその方法について検討した結果を示す．

（A）ポリシーの一元化

今回の実装ではセキュリティポリシーを対象として検討してきたが，本来の PolicyComputing では，セキュリティのみでなくネットワーク運用時，あるいはネットワーク利用時の通信品質に関するポリシーなども含め統一した考え方で管理したい．

従来通信品質についても PolicyComputing で対応することを検討してきており，ここでの検討結果を

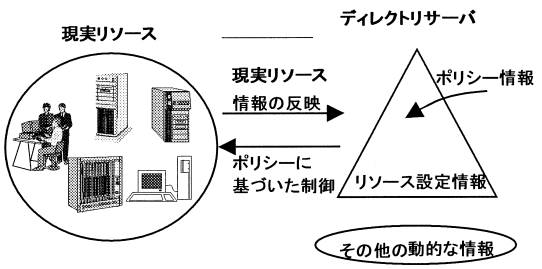

図 6 セキュリティポリシー情報の管理
Fig. 6 Management of the security policy information.

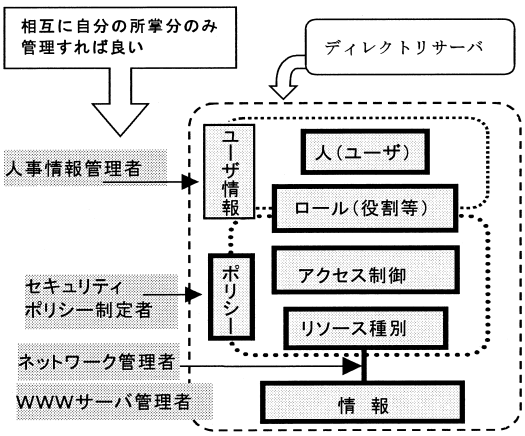

図 7 階層的ポリシー管理
Fig. 7 Hierarchical policy control.

ふまえセキュリティポリシーに関してもディレクトリサーバを利用し他のポリシーと統一的に扱い，2 章に示したメカニズムで管理していくこととする（**図 6**）．

（B）情報の一元管理

（1）で示したポリシー制御導入時のコスト増大を回避するには分散されたリソースの情報を把握していること，あるいは新しいリソースが追加されても問題なく管理情報を更新できることなどを柔軟にかつ拡張性をもって行うことが必要になる．

一方，実社会の組織構造は一般的に階層化されており，各階層ごとに管理者がいてその管理者は自分の所掌する範囲を管理しておけば全体的に管理ができるといった仕組みになっている．

今回は，このアナロジーによりポリシー情報についても階層化された構造により一元管理することを提案する．

ポリシーコンピューティングによりポリシーを集中管理する概念図を**図 7** に示す．

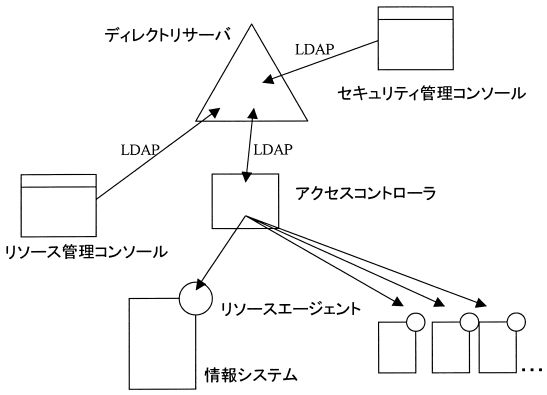

図 8 システムアーキテクチャ
Fig. 8 Overview of system architecture.

4. セキュリティのアクセス制御における PolicyComputing の実装と評価

4.1 実　　装

(1) アーキテクチャ

今回の実装では WWW サーバとして NETSCAPE ENTERPRISE SERVER，グループウェアサーバとして LOTUS 社の domino を用いた．

最初に，明文化されたポリシーをコンピュータが理解できる形式で電子的に保存する必要がある．

PolicyComputing ではそのレポジトリとしてディレクトリサーバを用いる．情報システムが存在するサーバに追加されるリソースエージェントはディレクトリサーバに蓄積されている情報を解釈し，個別の設定がそのポリシーを満たすように自動的な変更を行う．

また通常ディレクトリサーバは変更の通知を行うことができないため，アクセスコントローラがディレクトリの変更情報を各リソースエージェントに通知する．

情報作成者は直感的な設定が可能なユーザインタフェースにより，情報の種別を選択するのみでポリシーが適用される．

本システムのアーキテクチャを**図 8** に示す．

(2) アクセス制御ポリシーの変換

(1) で述べたようにアクセス制御ポリシーはディレクトリサーバのエントリとして表現される．

表 1 で示されるように，アクセス制御ポリシーは情報システムに依存しない記述方法で記載されるものであり，そのような抽象的な記述を定義できるようにする必要がある．

アクセスコントロールの仕組みは認証（Authentication）と許可（Authorization）に分類することができる．

許可のための設定は一般的にアクセスコントロールリスト（ACL）で記述され，あるネームスペース内で認証されたユーザもしくはロール（役割）ベースでグループ化されたユーザがあるリソースに対する対象行為をある条件下（時間など）で許可/拒否されるという形式で記述されている．個々の情報システムによりこれらの設定方法は異なるが，その形式は同様であるため，これらの情報をディレクトリのエントリとして定義することとする．

また，エントリのためのスキーマ定義を検討するにあたり以下のような点を考慮した．

① 同じ意味を持つ情報の冗長性の排除

たとえば［秘密扱い］ポリシーの内容を変更した場合，この［秘密扱い］ポリシーを表現する情報のみを変更すれば［秘密扱い］ポリシーを採用している情報のポリシーが変更されるようにする．

② 責任範囲に従ったアクセス制御の指定

ドキュメント管理者，システム管理者，セキュリティ管理者などが異なる際に管理範囲に応じて変更できるようにアクセス制御がなされる必要がある．また，ディレクトリの主な特性から以下の点も考慮する．

③ 変更頻度に応じたエントリ定義

一般にディレクトリは高速な検索が可能であるが更新処理に時間がかかる．そのため，更新頻度が異なる情報を同じエントリで定義することは全体的な性能低下をもたらすため避ける（たとえばポリシーの変更と人事異動の頻度は異なるため，ポリシーを示すエントリに人事異動ごとに変更を必要とするような情報を含めない）．

④ 関連付けの定義方式の検討

ディレクトリはその情報モデルからエントリの属性間の関係を示すことが難しい．そこでその関係を示すための情報の持ち方を検討する必要がある．

以上の検討課題を基にポリシー定義を行った．その内容を**図 9** に示す．

図 10 は検討の結果，表 1 で示されるようなポリシーを実際にエントリとして定義した例である．

ここに示すとおり，あるエントリがその属性値として他のエントリを指定するという形式でポリシーの定義を行っている．

このように定義することにより

① リソースの種類によらないポリシー定義が可能となり，将来的に様々なリソース（XML 文書など）が追加可能となる，

② ユーザ情報を一元的にディレクトリに統一することが可能となる，

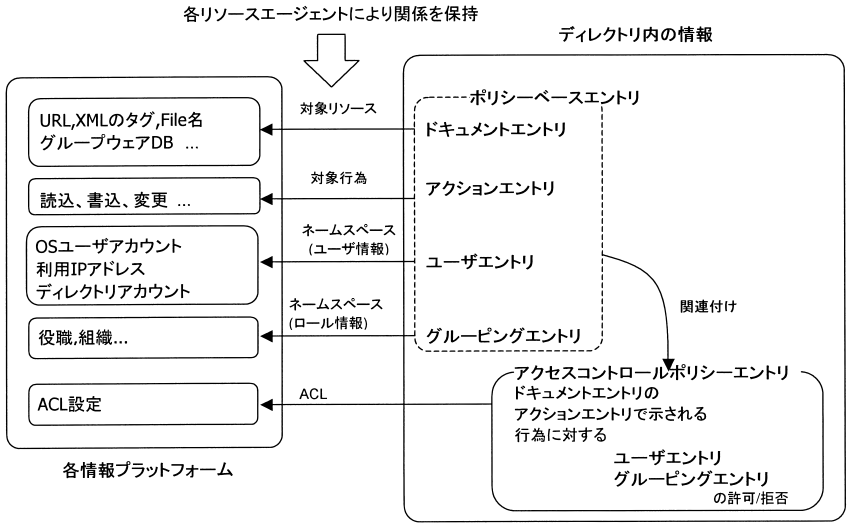

図 9 ディレクトリにおけるポリシーの定義
Fig. 9 Definition of the policy in the directory server.

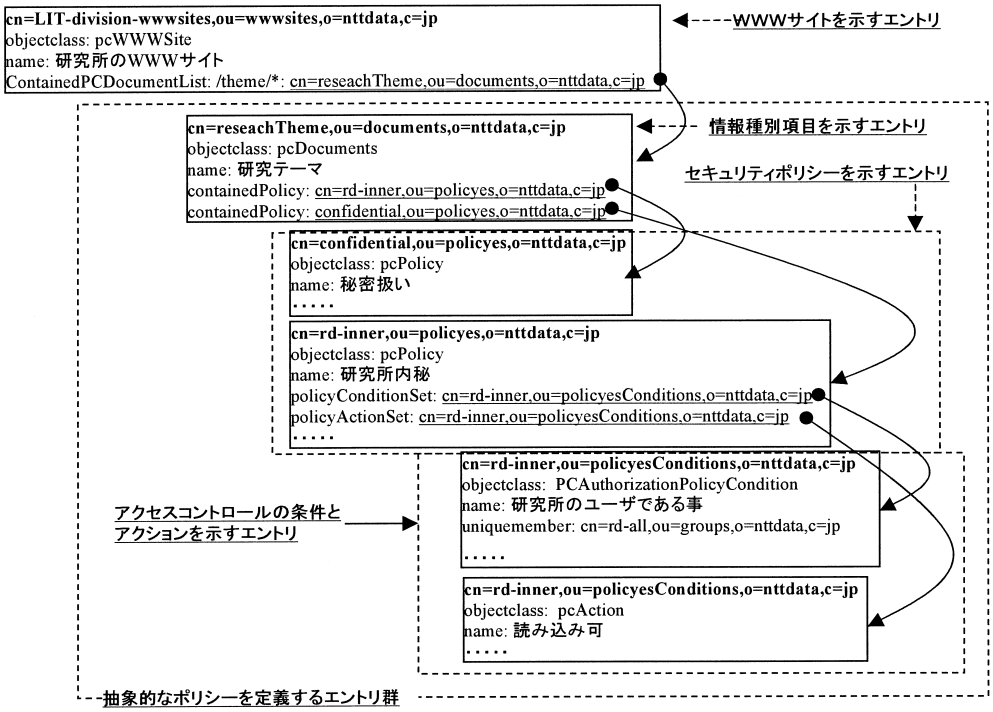

図 10 セキュリティポリシーエントリの例と関連する WWW サイトエントリの例
Fig. 10 Example of security policy entries and related www site entries.

③ アクセス制御ポリシーを管理する管理者自身は個々の情報システムの設定を考慮する必要がなく直感的な設定が可能である,

といった利点を得ることができる.

(3) リソースエージェントによる個別設定の変更

リソースエージェントはディレクトリサーバ上のポリシー情報を解釈し,個々の情報システムがそれに従い動作するように設定を変更するためのものである.

4.2 評 価

(1) システムの特長

4.1 節に示した仕組みを実現することによる本システムの特長は以下のとおりである.

ここでは，Web サーバを対象として管理する場合を示す．

（i）PolicyComputing によりセキュリティポリシーは一元化してディレクトリサーバに格納される．これにより情報セキュリティポリシーについてはこの情報だけを管理すればよい．また，これを行うことができるのはセキュリティ管理者のみである．

（ii）Web サーバ管理者は情報と情報種別を関連付けておき，後は情報の管理を行えばよい．

これによって管理者の所掌範囲が明確となり，相互の連携不足から生じるミスを削減することができるとともに，責任範囲も明確となる．

（2）管理コストの削減

情報作成者はアクセス制御ポリシー遵守のためにポリシーの解釈，設計/設定，監査/保守を行う必要がある．

この流れにおいて PolicyComputing の導入により情報の情報種別項目を選択するのみでその遵守が可能であるため，管理業務が削減される．

実際に削減できた管理業務の内容を以下に示す．

- 管理者は個々のドキュメントに対してアクセス制御を設定する必要がない．
- 管理者はユーザが増減したときにはグループへの登録削除だけを行えばよく，アクセス制御の設定を変更する必要がない．
- WWW サーバとグループウェアサーバで同じ設定を 2 回する必要がない．

一方，ツール操作の習得に時間がかかったなどのマイナス要因もあったが，これは新たなソフトウェアを利用するときの初期段階ではごく一般的に発生することであり運用が定常状態となれば解消できる問題である．

また PolicyComputing の導入により，管理業務の煩雑さのためにアクセス制御ポリシーから逸脱するような設定がなされているような状況が回避できることもセキュリティの向上をもたらすこととして期待できる．

5. おわりに

本論文では，ディレクトリサーバを用いてポリシーを一元管理する際の問題点である運用管理コストを削減する仕組みである「PolicyComputing」を提案し，通信品質制御やセキュリティ管理のアクセス制御に関して適用することが有効であることを示した．

また，実際に PolicyComputing を実装してアクセス制御に適用し運用管理コストが削減できることを確認した．

PolicyComputing の仕組みはマスターポリシーを定義することにより管理コストを削減することが特長であるが，各リソースの管理を行うための個別ポリシー間には矛盾が発生する可能性もありうる．

今後はこの矛盾を自動的に解決するための方式を検討していく．また，PolicyComputing を実商用システムに適用することにより実際の運用コストの削減程度を測定し有効性を検証していく

謝辞 本研究の機会を与えていただいた株式会社 NTT データ中村直司開発本部長，同荒川弘熙産業事業本部産業ビジネス推進本部長（前技術開発本部長）に深謝する．また日頃からご指導いただく同中村太一開発本部技術開発部長ならびに担当各位，および東北大学電気通信研究所木下哲男助教授に感謝の意を表する．

参考文献

1) 村田：マルチメディアコンピュータネットワークの通信品質保証，電子情報通信学会誌，Vol.81, No.4, pp.362–370 (1998).
2) 中野：「信頼できるインターネット」を実現する IP QoS とポリシー管理，情報処理，Vol.40, No.10, pp.1004–1006 (1999).
3) Judd, S. and Strassner, J.: Directory-enabled Networks, DMTF Draft (Sep. 1998).
4) Strassner, J. and Ellesson, E.: Terminology for describing network policy and services, Internet-Draft (Aug. 1998).
5) Croll, A. and Shivnan, A.: Policy-based networking and the role of directories, *3Com WhitePaper* (Mar. 1998).
6) 藤山，萱島，永井，角田，山田：セキュリティポリシー作成支援ツールの開発，情報処理学会研究報告，Vol.2000, No.30, pp.87–92 (2000).
7) 田中，菅野，小熊，松田：情報ネットワークシステムのポリシー制御 POLICYCOMPUTING に関する一検討，情報処理学会研究報告，Vol.99, No.18, pp.121–126 (1999).
8) 菅野，坂田，小熊，田中，白鳥：POLICY-COMPUTING のセキュリティポリシーへの適用，情報処理学会研究報告，Vol.99, No.56, pp.103–108 (1999).

(平成 12 年 5 月 29 日受付)
(平成 12 年 10 月 6 日採録)

菅野 政孝（正会員）

1950 年生．1976 年電気通信大学大学院修士課程修了．同年日本電信電話公社（現 NTT）横須賀電気通信研究所入所．1988 年以降，NTT データ通信〔現（株）NTT データ〕においてマルチメディア通信，インターネット，EC に関する技術開発，および IT 関連ビジネスモデルの開発に従事．現在英国支店長．著書「ネットワークセキュリティと暗号化」（共著，カットシステム）等．電子情報通信学会会員．

田中 俊介（正会員）

1972 年生．1997 年慶應義塾大学大学院理工学研究科修士課程修了．同年 NTT データ通信入社〔現（株）NTT データ〕．現在同社開発本部にて，ネットワーク技術の研究開発に従事．1997 年本学会全国大会奨励賞受賞．電子情報通信学会，ISOC 各会員．

坂田 祐司

1971 年生．1996 年東京大学大学院工学系研究科修士課程修了，同年 NTT データ通信〔現（株）NTT データ〕入社．現在同社開発本部にて，ウェブサービスアーキテクチャの研究開発に従事．

小熊慶一郎

1967 年生．1991 年早稲田大学法学部卒業．同年 NTT データ通信〔現（株）NTT データ〕入社．現在同社開発本部にて，セキュリティ技術の研究開発に従事．

白鳥 則郎（正会員）

1946 年生．1977 年東北大学大学院博士課程修了．1984 年同大学助教授（電気通信研究所）．1990 年同大学工学部情報工学科教授．1993 年同大学電気通信研究所教授．情報通信システム，ソフトウェア開発環境．ヒューマンインタフェースの研究に従事．1993 年本会マルチメディア通信と分散処理研究会主査．1985 年本会 25 周年記念論文賞受賞．本会理事（1996〜1998）．本会フェロー．IEEE Fellow．