修士論文　　2001年度(平成13年度)

環境情報を利用した情報家電のアクセス制御機構

慶應義塾大学大学院 政策・メディア研究科
氏名:楠本 晶彦

修士論文要旨　2001年度(平成13年度)

環境情報を利用した
情報家電のアクセス制御機構

本研究では，環境情報適応型アクセス制御モデルを提案する．また，環境情報適応型アクセス制御モデルを実現するKMSF-RUMシステムの設計，実装と評価について述べる．

近年，家庭内ネットワークの普及に伴い，情報家電という新しい家電が登場した．情報家電はネットワーク接続機能を持ち，遠隔制御や家電相互の協調動作が可能な点が特徴である．

パーソナルコンピュータやワークステーションといった既存の計算機環境と比較すると，情報家電環境は，動作するネットワーク，ハードウェアやソフトウェア，そして対象とするユーザ層において，ヘテロジニアス(heterogeneous)な環境であると言える．また，情報家電は誤操作や故障によって，身体に危険が及んだり所有物が破損する可能性がある点でも既存の計算機環境と異なる．

そのため，セキュリティを考慮する際にも，情報家電環境の特徴への配慮が必要となる．幅広いユーザ層が利用する情報家電セキュリティは，ユーザビリティを考慮しなくてはならない．また，運用上の問題や経験不足から生じる人為的ミスが人命に影響する可能性がある情報家電環境において，人為的ミスや誤作動に対する対処が必要である．

本研究では，以上のヘテロジニアスな情報家電環境において，人為的ミスや誤作動からのシステムとユーザの保護を目的とする環境情報適応型アクセス制御モデルを提案する．環境情報適応型アクセス制御モデルは，セキュリティポリシを環境情報を用いて表現する．また，個々のポリシを組み合わせて複雑な状況の記述を可能とする．これにより，ユーザは「場所」や「時間」など日常的に利用している方法で，セキュリティポリシを記述できる．

本論文では，情報家電環境の特徴とセキュリティについて関連研究を踏まえ考察し，環境情報適応型アクセス制御モデルを提案する．次に，KMSF-RUMシステムの設計，実装について説明し，定量的，定性的両評価を行う．本論文を通じて，KMSF-RUMシステムが実現する環境情報適応型アクセス制御モデルの有効性を実証する．

キーワード :1 情報家電　2 アクセス制御　3 環境情報　4 適応　5 Java

慶應義塾大学　政策・メディア研究科
楠本　晶彦

Environmental Information Adaptive Access Control Model for Networked Appliances

Abstract of Master Thesis

In this research, an access control model that adapt to environmental information is proposed. The design, implementation, and evaluation of KMSF-RUM system that satisfy the proposed model is also discussed.

In recent years, new kind of home appliances, or networked appliances, started to appear due to functional and performance development of home networks and home appliances. Networked appliances are equipped with network connectivity, and could be remotely controlled or cooperate with each other.

Compared with traditional computational environment based on personal computers and workstations, an environment based on networked appliances can be seen as heterogeneous. There is also a difference between these two environments in that with networked appliances, malfunctioning or failure may harm users physically, or damage their properties.

When considering security for networked appliances, their characteristics should be taken into account. Security for networked appliances with diverse users should contain high usability. Also, since human error in this environment could lead to physical risk of users, human error and malfunctioning must be appropriately met.

Adaptative access control model express security policy using environmental information. Also, several policies can be combined to express complex contexts. With these properties, users can enscript their security policy using daily expressions such as place and time.

In this paper, properties of networked appliance and its security are discussed in comparison with related research, and adaptative access control model is proposed. Next, design and implementation of KMSF-RUM system is explained, and both quantitative and qualitative evaluation of the system is shown. The effectiveness of the proposed adaptative model that KMSF-RUM utilizes is demonstrated through this paper.

Keywords : 1 Information Appliance 2 Access Cotrol
3 Environmental Information 4 Adaptive 5 Java

Keio University Graduate School of Media and Governace
Akihiko Kusumoto

目 次

図 目 次

表 目 次

第1章　序論

1.1　本研究の背景

近年，家庭内ネットワーク及び家電の高機能化や高性能化に伴い，情報家電という新しい家電が登場した．情報家電はネットワーク接続機能を持ち，遠隔制御や家電相互の協調動作が可能な点が特徴である．

パーソナルコンピュータやワークステーションといった既存の計算機環境と比較すると，情報家電環境はヘテロジニアス (heterogeneous) な環境であると言える．それは様々なプロトコルのネットワークを介して，異なるベンダが開発した，多種多様な機器が様々なユーザによって利用されるからである．

また，情報家電環境は物理的な動作を伴う．例えば，エアコンであれば室内の温度を変化させ，加熱調理器は燃焼を伴う．このような周囲に対する物理的な影響に対する配慮は既存の計算機環境において，あまり必要とされなかった．パーソナルコンピュータが暴走したからと言って，人的被害を被る可能性は低い．しかし，情報家電は誤操作や故障によって，身体に危険が及んだり所有物が破損する可能性がある．

このような情報家電環境において情報家電機器を安全に利用するためにはセキュリティが重要である．セキュリティに対する脅威を，表 1.1 のように整理する．

情報家電機器は主に，ホームゲートウェイなどで隔離されたホームネットワーク上で動作する．従って，悪意ある不正行為に対してはホームゲートウェイ上でファイアウォールなどの技術を用い，外部からのアクセスを制限する事は有用であり，一定の効果をあげている．

しかし，運用上の問題や経験不足から生じる人為的ミスが人命に影響する可能性がある情報家電環境において，人為的ミスや誤作動に対する対処が必要である．

表 1.1: 情報セキュリティのに対する脅威の構成

不正行為	不正アクセス コンピュータウィルス 機密情報漏洩 プライバシー侵害 著作権侵害 電子商取引に対する脅威 サイバーテロ その他の迷惑行為
人為ミス 誤動作	運用に係わる問題 知識不足から生じるミス 障害 2000 年問題
天災	地震，火事，水害

1.2 本研究の目的

本研究では環境情報適応型アクセス制御モデルを提案する．本モデルは意図しない脅威であるユーザの誤操作や故障に伴う損害の防止を目的としたアクセス制御を行う．情報家電環境下においてアクセス制御を行う場合，以下の点に留意しなくてはいけない．

まず，ユーザの直感的な設定や操作が必要である．情報家電環境においてユーザの操作に対する理解度は様々である．また，情報家電機器は日常的に頻繁に利用される．そのような環境下では位置情報や時間情報，温度情報などといった環境情報を用いて，ユーザの置かれる状況を指定可能である事が重要である．「母親と一緒」や「午前中」といったユーザが日常使っている指定方法でのアクセス制御が可能となる．

また，ヘテロジニアスな環境を吸収する事も情報家電環境におけるアクセス制御を行う上で重要である．上記のように，情報家電環境は異なる仕様やプロトコルが多く存在し，それらがネットワーク上に分散している．単一の情報家電でアクセス制御を行うだけでなく，システムとしてこれらの差を吸収したアクセス制御を行う事が必要である．

本論文では，**KMSF-RUM** システムを設計，実装した．KMSF-RUM システムは，環境情報を用いたアクセス制御機構を提供し，環境情報適応型アクセス制御モデルを実現する．

1.3 本論文の構成

本論文では，第2章において情報家電環境の特徴と情報家電環境におけるセキュリティについて触れ．環境情報適応型アクセス制御モデルを提案する．第3章では関連研究に触れながら，分散環境におけるセキュリティモデルやアクセス制御モデルについて論じる．第4章でKMSF-RUMの設計について，第5章で実装について述べる．そして，第6章でアプリケーションを提示し，第7章でKMSF-RUMの評価を行い，第8章で本論文をまとめる．本論文では以上のような構成をとり，環境情報適応型アクセス制御モデルとそれを実現するKMSF-RUMシステムについて論じる．

第2章　環境情報適応型アクセスコントロールモデル

本章ではまず，近年登場し始めた情報家電機器とその環境について，既存の計算機環境との対比を踏まえ論じる．

次に，情報家電環境のセキュリティモデルの特徴と要件について述べる．

最後に本論文で提案する「環境情報適応型アクセス制御モデル」の特徴と想定する環境を，シナリオを交えながら述べる．

2.1　情報家電環境

近年，パーソナルコンピュータ(PC)をはじめとする計算機はビジネス目的だけでなく，家庭にも急速に普及した．また，インターネット利用可能なユーザの数も大きく伸び，近年では家庭内からADSL(Asymmetric Digital Subscriber Line)や光ファイバーなどの高速なインターネットへの接続も可能になりつつある．今後，大容量の転送能力を利用したインターネットのサービスが家庭内へ普及していくことが予想される．

従来，PCを利用したインターネットへのアクセスが一般的であったが，現在はPCに限らず携帯電話や携帯端末などの機器からのインターネットへのアクセスが普及しつつある．図2.1に示すようにPCの出荷台数の伸びは鈍化しつつあり，時代は「ポストPC」時代に突入したといえる．

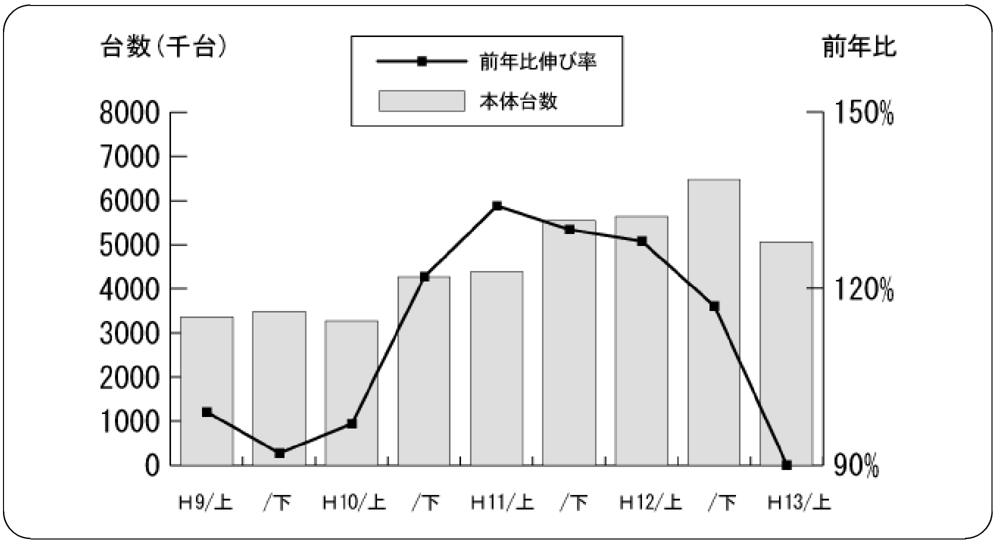


図2.1: PC出荷台数の推移

ポストPC環境では，従来のPCによるWWW(World Wide Web)の閲覧やメールの送受信だけでなく，多種多様な機器がネットワークに接続され，遠隔制御や協調動作が可能となる．

ポストPCの候補の一つが情報家電である．情報家電とはネットワーク接続機能を有し，遠隔制御や協調動作可能な家電である．例えば，インターネット接続機能を持った電子レンジ，携帯電話から遠隔制御可能な空調機器，ネットワークを介して相互に接続されるテレビやVCRが現在開発されている．単体で機能を果たしてきた家電がネットワークで相互に接続される事により付加価値の高く複雑な機能を提供できる．

情報家電環境とPCをはじめとする既存の計算機環境との相違点について論じる．

2.1.1 情報家電環境のユーザ

家電の特徴一つとして，利用するユーザ層の広さが挙げられる．テレビを例にとって考えると明らかであるが，テレビはお年寄りから子供まで幅広いユーザが利用している．こうしたユーザ層の広さは機器制御のための知識や能力の差を生んでいる．一般的にPCの利用は多くの知識と経験が必要とされる．これに対して，前述したように家電機器は幅広いユーザが利用する．応用的利用が可能なユーザから，単純な機能しか利用できないユーザまで様々である．情報家電に高度な設定や応用的な利用を要求するユーザがいる反面，VCRの予約録画が出来ないユーザが多く存在する事も事実である．

家電は誰にでも簡便に利用できなくてはならない．しかし，単に単純な機能を提供するだけでは，より高度な利用を要求するユーザは満足させられない．従って，複雑な機能を簡便に利用できる必要がある．さらに，家電がネットワークに接続される事で，情報家電の提供する機能は高度になり，操作はより複雑なものになる．

こうした高度な機能と複雑な操作を同時に実現するためには，機器の制御に対する環境情報の利用が有益である．例えばユーザの位置情報[1][2]や時間[3]，各種センサ情報を機器に取り込み，それらを機器の制御に反映させることが考えられる[4]．従来は，ユーザが自ら環境の情報を感知し，その環境に応じた操作を行っていた．しかし情報家電機器の利用や設定に，センサで取得した環境情報を利用することで，ユーザの経験不足を補い，ユーザは煩雑な操作や複雑な設定から開放される．

2.1.2 情報家電環境の計算機資源

情報家電環境は，様々なプラットフォームの上で動作する多種多様なソフトウェアが，異なるネットワークを介して接続される．現在，情報家電のネットワークインフラとしては，次のようなものがある．

- Ethernet (IEEE 802.3)
- IEEE 1394 [5]

- IEEE 802.11b/IEEE 802.11a
- Bluetooth [6]

上記以外にも情報家電同士を接続する仕様として，現在，赤外線通信やシリアル通信などが利用されている．オペレーティングシステム(OS)に関しては，WindowsやLinux，その他独自のOSなど，様々なOS上で情報家電ソフトウェアは動作している．現在，情報家電機器間を相互に接続し，協調動作機能を提供するミドルウェアも開発されている．主の情報家電を対象にしたミドルウェアを以下に示す．

- Jini [7]
- HAVi [8]
- UPnP [9] [10]

ソフトウェアやハードウェアに求められる要件に関して，情報家電機器と既存のPCとの大きな相違点として，低管理コストで運用できなくてはいけない点が挙げられる．PCの場合，OSやソフトウェアに障害や破損が生じた場合，最悪のケースではユーザはOSやソフトウェアの再インストールを要求される．これに対し，情報家電では，このようなOSやソフトウェアの再インストールをユーザに強いるのは非現実的である．また，PCやワークステーションなどの電源を切る際の煩雑な操作を要求する事は困難である．情報家電機器を低管理コストで運用するために，情報家電機器が，機器自体のハードウェアとソフトウェアを正常な状態に保つ必要がある．

2.1.3 情報家電の利用環境

現在，PCをはじめとする計算機の主な利用目的は，メールの送受信やWebブラウジング，一般業務から最近ではDV (Digital Video)の編集など，エンターテインメントへの利用まで拡大を続けている．しかし，ユーザの利用形態という観点では，ディスプレイの前に座り，キーボードとマウスで制御を行うといった限定的なものである．一方，情報家電の場合，利用目的はエンターテインメントから家事，遠隔コミュニケーションなど多岐及ぶ．加えて，ユーザは日常の様々な状況において情報家電を利用する．同じ機器を利用する際でも状況に応じて利用方法や目的が異なる．例えば，居間でくつろぎながら家族でAV機器を利用する場合と，調

理を行いながら利用する場合とでは利用目的や利用可能なユーザインタフェースが異なる．この様に情報家電利用環境において，ユーザの置かれる状況に応じたサービス提供が必要となる．

2.1.4 混在する環境への適応

情報家電環境は様々な側面においてヘテロジニアスな環境と言える．

- ヘテロジニアスな機器やソフトウェア
- ヘテロジニアスな利用環境
- ヘテロジニアスな習熟度のユーザ

これらのヘテロジニアスな環境は，情報家電の利用や管理はもちろんの事，家電開発の観点からも大きな障壁となっている．ヘテロジニアスな環境に適応できない家電機器は限定的な利用のみが可能であり，個々の家電で複数の環境に対応するためにはより多くの開発コストを必要とする．

情報家電のヘテロジニアスな環境が生み出す様々な問題を解決するためには，何らかの方法で情報家電機器を環境に適応的に動作させる必要がある．その一つの手段として，ネットワークシステムを用いたアプローチ [11] が存在する．情報家電機器をネットワークで相互に接続し，環境情報をはじめとする様々な情報や機器を共有する事で，管理コストや開発コストを低く抑えられる．

2.2 情報家電とセキュリティモデル

本節では，キュリティモデルについて論じ，情報家電環境におけるセキュリティモデルの特性や満たすべき要件を考察する．

2.2.1 情報セキュリティに対する脅威

情報セキュリティに対する脅威は，情報処理振興事業協会 (IPA) の報告書 [12] では図2.2の様に分類されている．これによると不正行為と人為ミスや誤作動，天災に大別できる．

不正アクセスやサイバテロなどの不正行為は近年深刻さを増している．しかし，情報家電環境では，これらの不正行為に対策はもちろんのこと，経験不足から生じるミスや運用に係わる問題への対策が重要である．

前述の様に情報家電はパーソナルコンピュータなどと比較して，事故によって人命に影響が出る可能性が高い．また，機器利用に関する理解度も様々である．この様な状況下では，ミスや故障などの意図的しない障害から生じる脅威への対策が，より重要である．

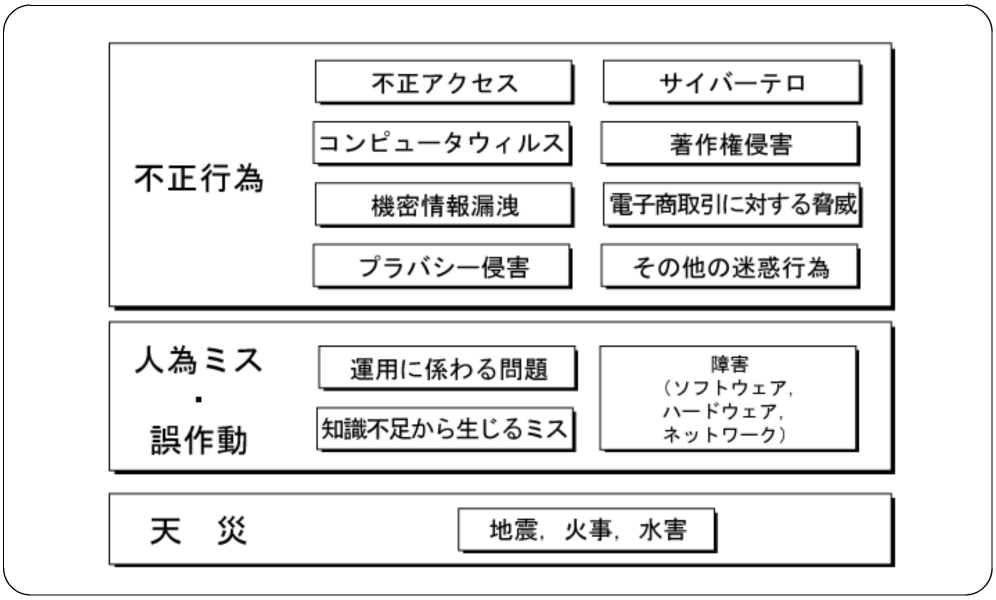


図 2.2: 情報セキュリティに対する脅威の構成

不正行為に対処する方法として，ホームゲートウエイなどの家庭内ネットワーク内と外の境界でファイアウォール [13] などの，外部からアクセスを制限する技術がある．これらの技術は基本的にはインターネット上で開発及び運用されているファイアウォールの技術を情報家電環境に適用したものである．

しかし，ユーザの誤操作や資源の競合などからシステムを保護する技術は少ない．複数の情報家電機器が家庭内ネットワーク上で機能を共有を行う状況は操作や資源の依存関係の複雑化を生む．これにより，誤操作や競合の危険性がより一層高まる．これらの問題を個々の家電機器内や OS 内で解決する事は困難である．したがって，接続先と接続元を含め，情報家電システム全体で保全性に対する取り組みを行う事が重要である．

2.2.2 ユーザビリティとセキュリティ

銀行口座などの個人情報や企業の機密情報をネットワーク上で扱う場合には信頼性の高いセキュリティが必要となる．例えば，インターネット上で銀行のオンライン振込みや残高照会のサービスを利用する場合，IDとパスワードを利用した認証をユーザは要求される．一般にセキュリティの強度とユーザビリティは相反の関係にある．強固なセキュリティ機能をシステムに対して実装した場合，ユーザは煩雑な操作や複雑な設定を要求される．また，どういったセキュリティポリシやモデルを適用するかを決定する際にも，適用対象の用途や利用形態についても考慮する必要がある．

日常的に頻繁に利用する家電機器において，ユーザに煩雑な認証操作を要求する事は困難である．また，情報家電環境では重要な機密情報などを扱わない機器や家電ネットワーク外部と通信しない機器が多い．このような情報家電ネットワークにおいて強固なセキュリティ機構を提供する事よりも，ユーザビリティを下げないセキュリティ機構を提供する事の方が重要である．また，機器や機能によって扱う情報の重要度が異なる事から，機器や機能に応じたセキュリティポリシやモデルを選択し，導入できる事が重要である．

ユーザビリティを下げずに認証機能をする方法として，指紋や光彩等のバイオメトリクスやJava Cardなどのデバイスを利用することも考えられる．これらの認証方法や家電機器の認証においては有益であり，ユーザはキーボードやボタンによるパスワードの入力といった煩雑な操作から開放される．

2.2.3 静的なモデル対動的なモデル

セキュリティポリシは，静的なモデルと動的なモデルに大別できる．静的なモデルは固定的で決まったポリシに基づいたセキュリティ機構を提供する．一見するとセキュリティポリシは静的であると思えるが，実際は動的である．例えば企業内で役職に応じてアクセス権限を与えるような場合，従業員の配置転換によって許可される権限も変化する．

このような動的なセキュリティポリシの変化は情報家電ネットワークにおいても起こりうる．情報家電ネットワークでは複数のユーザが異なる機器を様々な状況で使い分ける事が想定される．この様な環境下ではユーザや機器，用途，状況に応じて動的にセキュリティポリシを変更し，

適応させる必要がある．

2.3 環境情報適応型アクセス制御モデル

前述したように，情報家電環境のセキュリティには多くの特徴がある．本論文では，環境情報適応型アクセス制御モデルを提案する．本モデルは特に情報家電環境の保全性向上を目的とする．

本モデルでは，ヘテロジニアスなユーザ層，ヘテロジニアスな機器やネットワーク，そしてヘテロジニアスな利用環境を対象とする．

従来のアクセス制御モデル多くは，PCやワークステーションなどの利用を想定し，業務用途を対象としていた．そのため，アクセス制御は厳密さが追求され，アプリケーション依存なアクセス制御が中心であった．従って，ユーザビリティへの配慮は乏しく，多様な機器や利用環境への柔軟な適応はできない．

本モデルでは環境情報を利用することで，ユーザは直感的指定でアクセス制御ポリシの設定や変更が可能となる．また，アクセス制御ポリシ細分化し，組み合わせる事で，ポリシの柔軟な表現や動的な変更が可能となる．

表2.1に環境情報適応型アクセス制御モデルの特徴を示す．

表 2.1: 環境情報適応型アクセス制御モデルの特徴

	環境情報適応型アクセス制御モデル	既存のアクセス制御モデル
目的	簡便性，柔軟性	高信頼性
適用範囲	情報家電環境	業務環境
動作環境	ヘテロジニアスな環境	閉じた単一システム内
管理者	一般ユーザ	専門知識のある管理者

2.3.1 保全性

保全性は外部からの意図的で悪意のある攻撃に対するものと，計算機資源の競合やユーザ誤操作などの意図しない脅威に対するものがある．本モデルにおいては，操作ミスや故障などの意図しない障害から生じる脅威に対する保全性向上を目的とする．

既存の計算機環境では，計算機資源の競合や制御衝突によって機器の正常な動作が損なわれたり，ユーザの誤操作によって計算機資源が損傷を受ける事に対する保全への取り組みが行われてきた．情報家電環境では，ユーザによるソフトウェアやアプリケーションのインストールが困難であるため，計算機資源を一定の状態で維持する事は非常に重要である．従って，本モデルでも計算機資源の管理を通じて，計算機資源を一定の状態に維持するといった保全を行う．

加えて情報家電機器の場合，PCを始めとする従来の計算機とは異なり，機器の誤操作や故障に伴い，ユーザが自身の怪我や所有物の破損などの物理的な損害をこうむる可能性がある．本モデルでは，この様な安全な機器利用を保証する保全性に関しても対象とする．

2.3.2　柔軟性

本モデルでは，情報家電環境の複雑で動的な利用環境に適応させるために，アクセス制御ポリシを組み合わせ可能とする．アクセス制御ポリシを組み合わせることで，機器や状況に応じた複雑なポリシ表現を可能とする．

また，組み合わせを動的に変更可能にする事で，刻一刻と変化する情報家電環境やユーザの要求に対応する．

2.3.3　ユーザビリティ

本モデルにおいて，アクセス制御の設定や組み合わせの変更は，ユーザにとって簡便でなくてはならない．幅広いユーザ層が利用する情報家電環境において，簡便な操作や設定は必須である．そのため本モデルでは，環境情報を利用して，アクセス制御の指定や設定を行う．それにより，ユーザは「場所」や「時間」などを指定する事で，アクセス制御のための状況記述が可能になる．

2.3.4　シナリオ

環境情報適応型アクセス制御モデルを用いると，次のようなシナリオを実現できる．

シナリオ1

VCRの録画中にTVからVCRの再生機能を利用する場合を考える．この様にある機能が利用されている際に，他の機能の利用を制限したり，禁止したい場合が存在する．機器の境界を越えた機能の共有を行う際には，複数の機器にまたがる排他制御の機能が必要である．

また，機能によっては多重を許す場合がる．例えば，ハードディスクレコーダの読み取り機能などでは，ある程度の多重を許す．しかし，動作の品質を保証するために，一定の多重度でアクセスを制限したい場合がある．この様な多重度を考慮したアクセス制御を行う事は，家電メーカが想定した動作環境に近い環境で機器を利用できる．

シナリオ2

家事の際に調理器具を利用を考える．換気扇が動作してない場合，調理器具の電源が入らないようなアクセス制御を行う．また，これらの機器と一酸化炭素濃度を測定するセンサなどを組み合わせる事も考えられる．このような機器同士の依存関係や，センサ情報に基づいたアクセス制御を機器間を越えて行う事で，火災や中毒などの事故の発生を未然に防げる．

シナリオ3

例えば，母親が子供のTV使用を深夜に限って制限したい場合がある．この様に，時間帯によって機器や機能の利用禁止や警告が可能になる事で，ユーザは直感的に利用制限の指定が出来る．

また，機器やユーザの位置情報を利用制限に用いる事も有益である．保護者が同じ部屋にいるときにのみ利用可能なコンロや3人以上部屋にいる時のみ利用可能なエアコンの暖房機能が考えられる．

ユーザが状況を直感的に指定できるといった意味で，時間や位置の情報や有益である．これらの情報や他の環境の情報を組み合わせて利用することで，より直感的な状況の指定が容易になる．例えば，午後10時以降は親の同伴なしでは利用することが出来ないTVなども考えられる．

2.4　本章のまとめ

本章では情報家電環境の特徴について述べ，PC を始めとする既存の計算機環境と比較した．

次に情報家電環境下でのセキュリティモデルについて論じた．セキュリティモデルについても情報家電環境特有の問題点を指摘し，満たすべき要件について述べた．

最後に本論文で提案する「環境情報適応型アクセス制御モデル」について述べ，特徴や想定する環境について触れた．また，3 つのシナリオを通じ環境情報適応型アクセス制御モデルが目指す利用例を示した．

第3章　関連研究

本章では，代表的なアクセス制御モデルとその分類について述べる．

また，分散システムにおけるアクセス制御やJavaにおけるセキュリティについて触れ，情報家電環境におけるアクセス制御を考察する．

3.1 アクセス制御

本節では情報家電環境のセキュリティに重要な役割を果たす，アクセス制御モデルについて考察する．アクセス制御とは，ユーザが計算機資源にアクセスする事のできる権限・認可を制御することを指す．

アクセス制御はDAC(Discretionary Access Control)方式とMAC(Mandatory Access Control)方式に大別できる．

本節では，ユーザやプロセスなどをサブジェクトと呼び，ファイルやデータレコードなどの資源をオブジェクトと呼ぶ．

3.1.1 適応型アクセス制御モデル

アクセス制御モデルは，適応型アクセス制御モデルと非適応型アクセス制御モデルに分類できる．従来のアクセス制御モデルの大半は非適応型アクセス制御モデルである．その理由として，従来のアクセス制御モデルの多くは業務目的に開発され，高信頼なアクセス制御を必要とし，柔軟な状況の変化に適応する必要が無かったからである．厳密なアクセス制御を管理，運用する上で頻繁に状況が変化する環境は望ましくない．

また，適応型アクセス制御モデルは，付与されている権限が適応的に変化するモデルとアクセス制御ポリシ自体が適応的に変化するモデルに分類できる．前述したように，意識的に厳密なアクセス制御のポリシ管理を行う上で，アクセス制御ポリシ自体が適応的に変化するモデルは好ましくない．しかし，情報家電環境の様に，アクセス制御が行われている事を強く意識したくない環境では，アクセス制御ポリシ自体が状況の変化に適応的に動作する事が重要である．

第2章で提案した環境情報適応型アクセス制御モデルでは，アクセス制御ポリシ自体が環境情報へ動的適応をする．

3.1.2 Mandatory Access Control

MAC方式のアクセス制御は，必須アクセス制御と呼ばれ，軍事機密などの機密管理に利用される．

MACでは，これらのサブジェクトとオブジェクトの双方に順位付けされた機密ラベルを付加する．機密ラベルは最高機密(Top Secret)，機密(Confidential)や機密対象外(Unclassified)などの区分レベルで階層的に記述される．サブジェクトの機密ラベルとオブジェクトの機密ラベルを

比較し，一定の MAC チェックにパスしない場合はアクセスを拒否する．MAC が必須である理由は情報のラベル付けが自動的に行われ，通常のユーザーは管理者から承認を得ていないかぎりラベルを変更できないためである．

MAC では，アクセス許可決定のロジックは機密ラベルの区分体系に組み込まれているため，区分レベルの階層を定義するだけでアクセス制御の管理が可能であり，ACL(Access Control List) の維持，管理の労力を軽減できる．

3.1.3 Discretionary Access Control

DAC 方式のアクセス制御は，任意アクセス制御と呼ばれる．

DAC では，UNIX のファイルシステムにおける制御のように，個々のユーザやグループに対してアクセス権または特権を明示的に与える．

3.1.4 Role-Based Access Control

RBAC(Role-Based Access Control) では，ユーザやユーザグループに対してロールと呼ばれる役割を割り当てる．オブジェクトはロールごとのアクセス権限を記述する．

RBAC では，実世界で用いられる職場での役職などをそのままアクセス制御に利用できる．例えば，「人事部員」などといったロールがユーザに割り当てられ，ファイルなどのオブジェクトもロールごとのアクセス制御がなされる．従って，「人事部員のみが閲覧可能なファイル」などの指定が可能になる．

3.2 分散環境とセキュリティ

情報家電機器は，家庭内ネットワークを介して遠隔制御や協調動作することが想定される．この様な遠隔制御や分散協調動作を構築し，開発を支援する技術として分散オブジェクトが挙げられる．

本章では既存の分散オブジェクト環境に触れ，その分散オブジェクト環境内のセキュリティについて考察する．分散オブジェクト環境として Jini，CORBA(Common Object Request Broker Architecture)[14] そして EJB(Enterprise JavaDeans)[15] について触れる．

3.2.1 Jini

Jiniは，Sun Microsystems社が提唱する分散システムで，リソースとそのユーザの連合体を形成することを目的としている．

具体的にはルックアップサービス，サービスプロバイダ，クライアントの3者によってJini連合体は形成される．これら3者を用いてJiniでは，分散システムを通じて，リソースの登録，提供，検索や利用が行われる．

サービスプロバイダはハードウェアやソフトウェアリソースをネットワーク上にサービスとして提供する．サービスプロバイダによって提供されたサービスをクライアントが利用する．サービスの登録や検索はルックアップサービスを用いて行われる．図3.1にJiniが想定する利用モデルを示す．

Jiniを用いる事で，ネットワーク上のサービスを共有し，利用するシステムを容易に構築できる．

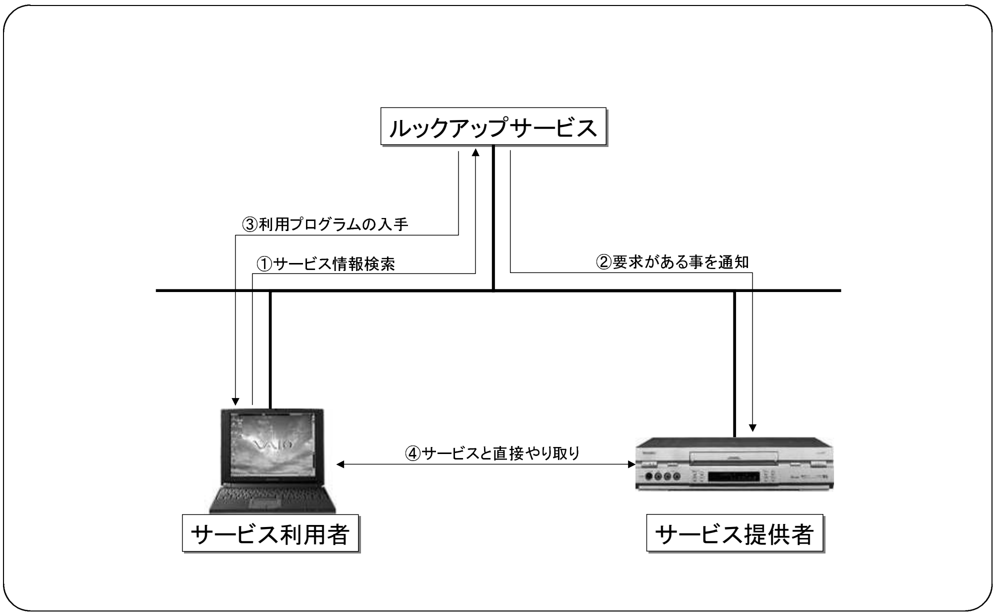


図3.1: Jiniの利用モデル

Jiniのセキュリティ

Jiniシステムでは，ネットワーク上の分散プロセス間で大量のモバイルコードが交換される．また，モバイルコードがサービスプロバイダの

リソースにアクセスする場合もある．この様なJiniの分散オブジェクト環境では，必然的にセキュリティが重要となる．

Jiniのセキュリティモデルは基本的にJava2のセキュリティメカニズムから派生したものであり，RMIセキュリティマネージャを使用している．

Jiniでは，サービスプロバイダやクライアントがルックアップサービスを検出する際にグループと呼ばれる文字列型の識別子を指定する．指定されたグループ以外のアクセスを禁止することで，ルックアップサービスは他のグループから登録されているサービスプロバイダの情報を隠蔽できる．

しかし現状として，Jiniセキュリティの対象範囲は非常に限定されたものである．ルックアップサービスが実際にそのグループに登録されているかどうかを保証するメカニズムは無い．

ユーザ認証の機能もJiniでは提供されていない．ネットワーク上で安全にサービスを享受を行うには，JAAS(Java Authentication and Authorization Service)[16]のような統一されたユーザ認証のメカニズムが必要である．

3.2.2 EJB

EJBはサーバ側のエンタープライズアプリケーションコンポーネントのためのモデルを提供する．EJBではJavaBeansと呼ばれるコンポーネントを相互に接続し，利用することで再利用性の高いプログラミングモデルを実現する．このモデルを用いる事で，移植可能な分散型のエンタープライズアプリケーションをスタンドアロンのエンタープライズアプリケーションと比べて，容易に構築できる．

EJBではEJBコンテナやEJBサーバ内でEJBコンポーネントが実行される．EJBコンテナやEJBサーバにはEJBコンポーネントにおける分散通信サービス，データイネーブリングサービス，システム保証サービスが提供されいる．これにより，少量のプログラミング量でアプリケーションの開発が可能となる．

EJBのセキュリティ

EJBコンポーネントはEJBコンテナ環境内で動作している．したがって，EJBインスタンスの生成や破棄，EJBインスタンスのオン／オフの切り替え，EJBのライフサイクルの一般的な管理などはEJBコンテナ仕

様に依存する．

EJB のセキュリティメカニズムも EJB 標準のメカニズム，EJB コンテナに依存したベンダによる独自のメカニズム，EJB コンポーネント開発者が独自にコーディングするメカニズムがある．

図 3.2 に EJB のセキュリティアーキテクチャを示す．標準セキュリティメカニズムは，ロールベースのアクセス制御に必要な最小限の機能を提供する．しかし，分散環境でのユーザをこのロールに対応付けるための方法は規定されていない．このため，ネットワーク上でのユーザ ID の伝播や委譲，認証は EJB コンテナのベンダに委ねられている．

また，既存の EJB コンテナが提供するセキュリティポリシは XML を始めとした静的なファイルが中心である．

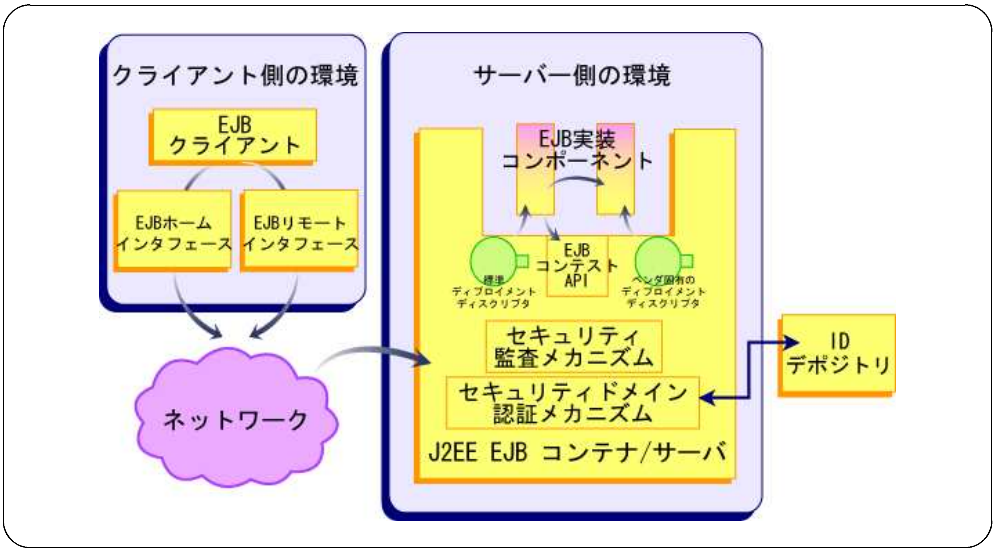


図 3.2: EJB のセキュリティアーキテクチャ

3.3 Java言語のセキュリティ

本章では，Java言語に関連するセキュリティについて触れる．

3.3.1 Java 2 セキュリティ

JDK1.2 セキュリティアーキテクチャ[17] は，以下のような JDK1.0 や JDK1.1 の問題点への取り組みがなされた．

- アプレットに対する極端なサンドボックスの制約
- ポリシと施行と間の不十分な分離
- 脆弱な内部セキュリティメカニズム

JDK1.2 セキュリティアーキテクチャは，拡張性とスケーラビリティを持つ柔軟なアクセス制御を目的としている．しかし，ユーザ認証に対する統一的な手法が無く，ユーザ情報に基づいたアクセス制御が不可能である．また，基本的には設定ファイルに記述されたポリシに基づきアクセス制御を行うため，アクセス制御ポリシを動的に変更する事は困難である．

3.3.2 JAAS

JAAS は，J2SE V1.3 で具体化された Java 2 セキュリティモデルの標準拡張機能である．認証されるユーザID に基づいてリソースへのアクセスを制限するための標準的な方法を提供する．また，JAAS はログインやログアウトの処理を行うための標準的な機構も提供している．

JAAS の認証処理は認証方法や認証アルゴリズム提供する認証サービスを実装から切り離すため，PAM (Pluggable Authentication Module) モデル [18] を採用している．図 3.3 に PAM の概念図を示す．PAM ではコンフィグレーションによってログインモジュールをロードする．ログインモジュールはユーザを一意に認証する．コンフィグレーションの記述は図 3.4 に示すように，複数のログインモジュールを指定する事も可能である．複数のログインモジュールを指定した場合，複数の認証が連続して行われる．これによって，スマートカード認証後に指紋認証を行うといったことが可能となる．

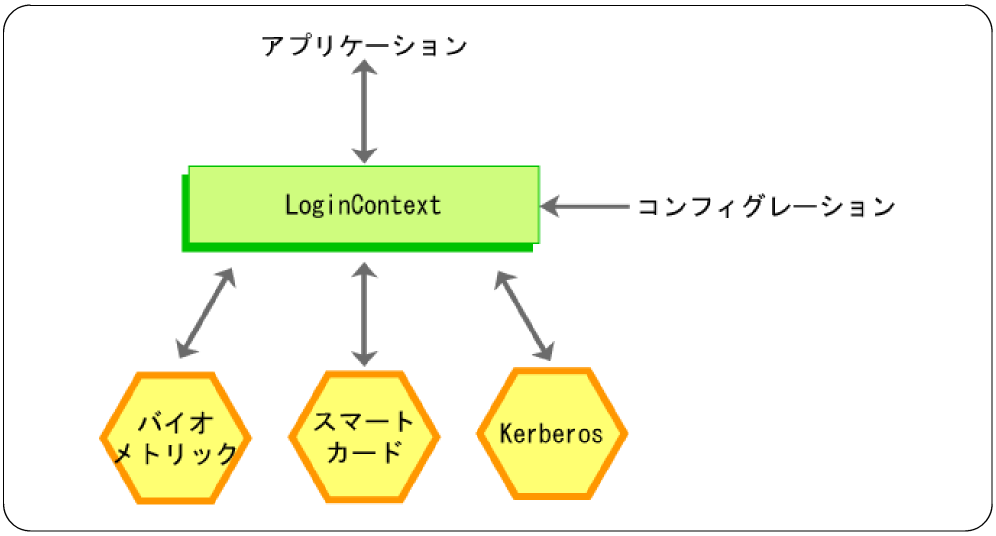


図 3.3: PAM の概念図

```
Login{
   com.sun.security.auth.SampleLoginModule REQUIRED debug=true;
   com.sun.security.auth.SolarisLoginModule REQUIRED;
}
```

図 3.4: JAAS のコンフィグレーション

上記の様に，JAAS はユーザ認証と認証後のアクセス制御機構を提供する．しかし，標準 JAAS でのセキュリティポリシは Java2 セキュリティと同様に静的なファイルで表現される．このため，セキュリティポリシを動的に変化させることは困難である．

3.4 本章のまとめ

本章では，関連研究として既存のセキュリティモデルと分散システムや Java におけるセキュリティの現状について触れた．

しかし，既存のセキュリティモデルやシステムはビジネスで利用されることを想定されて開発されたものが中心である．業務で利用されるセキュリティポリシやメカニズムは頻繁に変更される事を想定していない．

したがって，第 2 章で提案した環境情報適応型アクセス制御モデルの

ように，情報家電環境で想定される様々な利用環境を想定し，セキュリティポリシやメカニズムを状況に応じて柔軟に変更する事は困難である．

また，ユーザビリティに対する配慮も不十分であり，アクセス制御の設定や運用は専門的な知識が必要とされるため，情報家電環境に容易に適応できない．

第4章　KMSF-RUMシステムの設計

本章では，まず情報家電適応型アクセス制御モデルを実現するKMSF-RUMシステムの概要について述べる．

次に，KMSF-RUMの必要機能やシステム構成と動作手順について触れる．

最後にKMSF-RUMシステムを構成する通信部，認証部，アクセス制御部，名前解決部の詳細設計について述べる．

4.1　RUM システムの概要

環境情報適応型アクセス制御モデルを実現するために，本論文では **KMSF-RUM** システムを構築した．以下で KMSF-RUM システムの詳細と本システムが対象とする情報家電環境について述べる．

4.1.1　KMSF-RUM

KMSF-RUM システムは，環境情報を利用しユーザが直感的に理解しやすいアクセス制御機構を提供する．これにより，第 2 章で述べたような保全性の高い情報家電環境を提供する．

本システムはサーバ機能とライブラリ群に大別できる．サーバ機能はアクセス制御や認証，環境情報の管理といった機能を提供する．また，本システムでは情報家電開発者が容易に環境情報適応型アクセス制御を各情報家電機器上に実装するためのライブラリ群を提供する．

4.1.2　対象環境

本システムは，理解度の異なるユーザが混在する情報家電利用環境を対象とする．また，異なるメーカが情報家電機器ネットワークを通じ，機器の境界をこえて機能を相互に共有する環境を想定する．情報家電ベンダが容易に環境情報適応型アクセス制御の実装やアクセス制御ポリシの記述が行える事を目指す．

本システムは，ホームゲートウェイなどで区切られた家庭内ネットワーク上での動作を想定している．

4.2　必要機能

環境情報適応型アクセス制御モデルを実現する上で，以下の項目が本システムに必要な機能であると言える．

認証と識別

　アクセス制御を家庭内ネットワーク上の情報家電機器に提供するためには，ユーザを認証し識別する必要がある．

　機器や機能，ユーザによって適切な認証方式が存在する．TV の利用するたびにユーザにパスワード入力を求める事は非現実的である．

従って，本システムでは機能やユーザごとに異なる認証方式を提供する．

また，認証方法はICカードの利用や指紋認証も考慮する必要がある．

ポリシの記述

本システムでは環境情報を用いた直感的なアクセス制御ポリシを目指す．同時に情報家電機器ベンダが効率的に環境情報を用いたアクセス制御を実装できなくてはならない．従って，本システムにおけるアクセスポリシは柔軟な記述力と再利用性を備えていることが重要である．

ポリシマッピング

本システムでは認証時に得られたユーザの情報や機能の情報を利用し，適切なアクセス制御ポリシを選択する必要がある．これにより，ユーザや機能に固有なアクセス制御を実現できる．

機能名の変換

本システムは選択されるアクセス制御ポリシの設定や環境情報のパラメータ設定をユーザに提供する．この様な設定は詳細に行えなくてはならない．しかし，ユーザは難解な資源名や実装に依存した機能名を理解できない．従って，「TVの電源を入れる」といったような指定方法で機能に関する設定を行いたい．

これら詳細な設定と簡易な指定の両方を実現するためには，資源名や機能名をユーザが理解できるような形に変換する必要がある．

4.3 システム構成

本システムは図4.1に示すようにRUMサーバと各情報家電上に存在するRUMライブラリから構成される．RUMサーバはホームゲートウェイ内などに実装する事を想定する．個々の情報家電機器内に組み込まれたRUMライブラリはRUMサーバと通信し，RUMサーバの提供する認証機能やアクセス制御機能を利用する．

RUMサーバ内の構成を図4.2に示す．RUMサーバは通信部と認証部，アクセス制御部，名前解決部の4つによって構成される．それぞれの機能の説明を以下に示す．

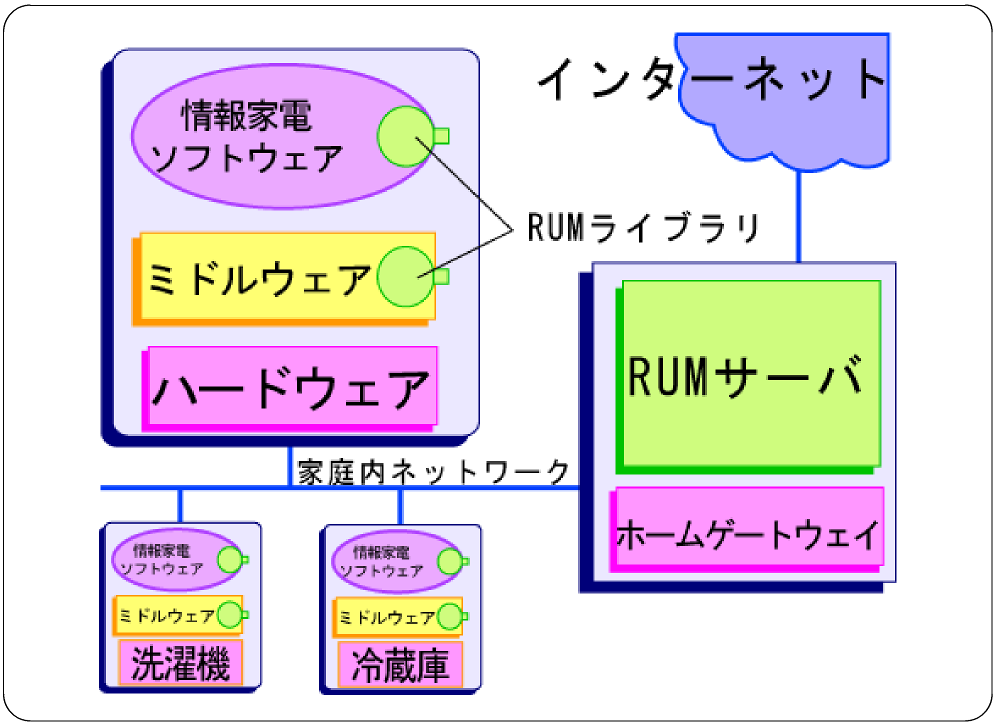


図 4.1: KMSF-RUM システムと情報家電の関係

通信部

各情報家電機器内の RUM ライブラリから認証やアクセス制御に関する問い合わせを受け，認証部やアクセス制御部に要求を出す．

認証部

通信部が RUM ライブラリから受けた認証要求を処理する．認証方法の選択を行い，選択された認証方法に従って認証の成否を通信部に返す．

アクセス制御部

RUM サーバの根幹を担う．ユーザと機器の識別子を通信部から受け取り，環境情報を利用したアクセス制御ポリシの評価を行い，結果を返す．また，評価結果に応じた動作を起動する．

名前解決部

認証部およびアクセス制御部に名前解決機能を提供する．ユーザが複雑な設定を容易に行えるよう，資源名や複雑な機能名をユーザが理解可能な形に変換する．変換された結果はユーザインタフェース

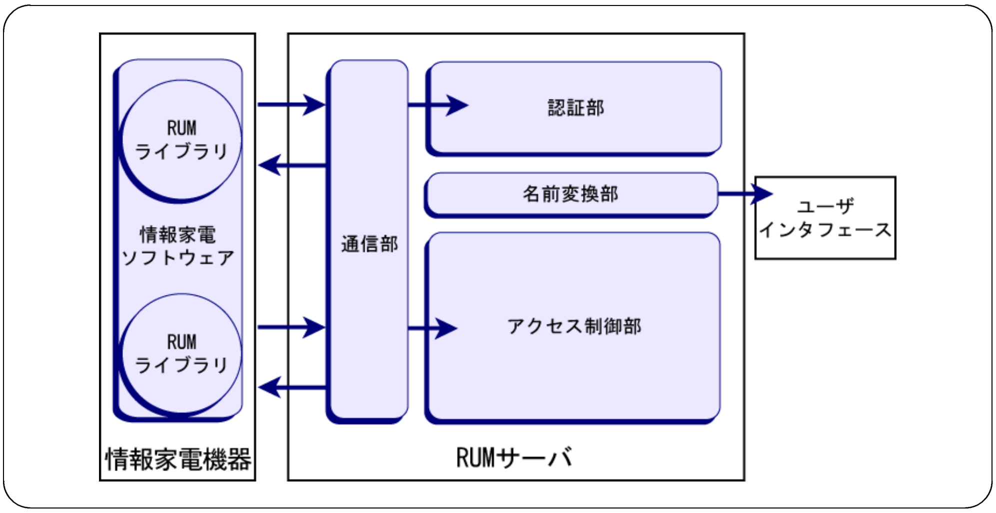


図 4.2: RUM サーバの構成

上などで利用される．

4.4 RUM システムの動作概要

本節では RUM システムの動作手順について述べる．図 4.3 のフローチャートに示されるように各情報家電機器上の RUM ライブラリから要求を受け，RUM サーバは認証済みか否かを判断する．認証が未了の場合は機能や機能の識別子から認証方法を検索し，RUM ライブラリを通じて認証をユーザに促す．

認証が完了するとユーザ識別子と機能識別子が与えられ，それらの識別子に応じたアクセス制御ポリシを検索する．検索結果のポリシに従い，環境情報を利用したアクセス制御の評価を行う．最終的に評価結果は RUM ライブラリを通して，各情報家電機器に返送される．

4.5 通信部の設計

本節では通信部の設計詳細について述べる．

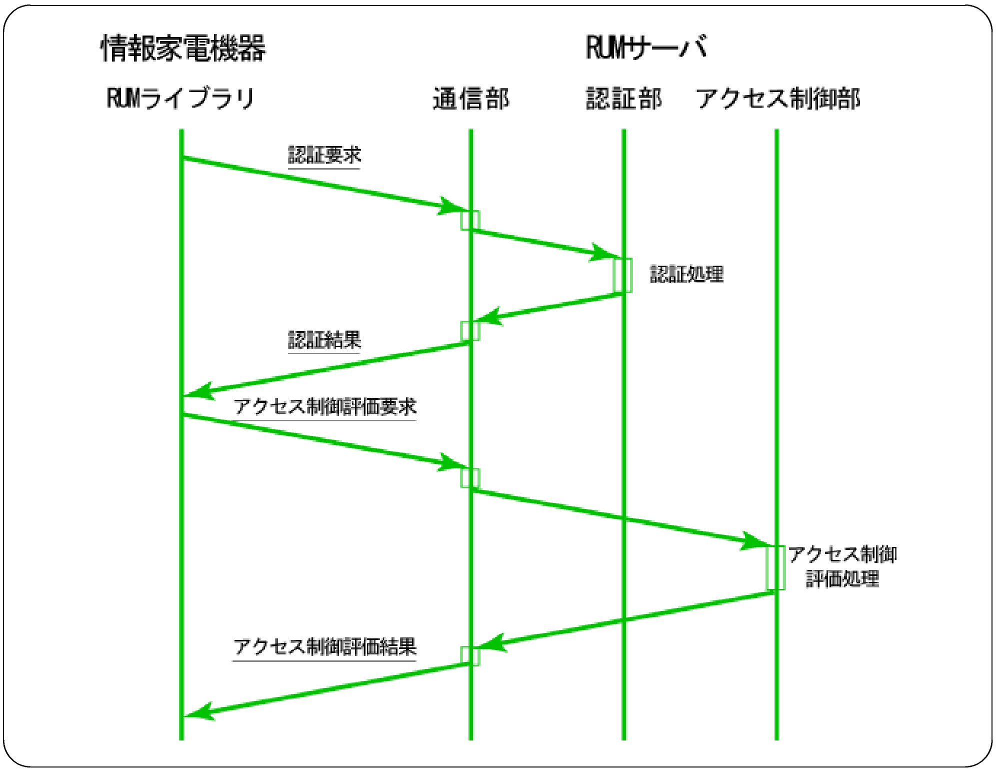


図 4.3: RUM システムの動作手順

4.5.1 RUM サーバへの入力

通信部は情報家電機器上で動作する RUM ライブラリからの通信を受け付ける．通信部は入力として表 4.1 に示すようなデータを受け取る．

これらの識別子は認証方法の決定やアクセス制御ポリシの検索・評価に利用される．

4.5.2 認証の必要性管理

通信部はトランザクションの認証状態を管理する．トランザクションの認証未了の場合は認証部に処理を渡し，認証が完了している場合は処理をアクセス制御部に渡す．

表 4.1: RUM サーバへの入力

入力項目	説明
ユーザ識別子	RUM ドメイン内でユーザの識別に利用
機能識別子	呼び出し元の機器や関数の識別に利用
トランザクション識別子	一連のトランザクションを識別
拡張	付加的な識別に利用

4.6　認証部の設計

本節では認証部の設定について述べる．

4.6.1　RUM ドメイン

本システムでは認証は個々の情報家電機器に対して行われるのではなく，RUM ドメイン単位で行われる．RUM ドメインとは1つのRUM サーバが管理するネットワークを指し，通常はホームゲートウェイなどによってインターネットから隔離された家庭内ネットワークである．ユーザはRUM サーバを通してこのRUM ドメインにログインし，1つのRUM ドメインにおいて一意に識別される．

4.6.2　PAM

情報家電機器やその機能ごとに適切な認証方法が存在する．例えば，TV の操作を行うためにパスワード認証を行う事は非現実的である．このように機器や機能ごとでの認証方法の変更を行うために本システムではPAM を採用した．

PAM はJAAS で利用され，プラグ可能でスタックされた認証を行う．このようにコンフィグレーションによって適切な認証方式を変更し，提供する．また，複数の認証を階層的に行う事も可能である．たとえば，スマートカードによる認証を行った後に，Kerberos[19] による認証を行うことが想定できる．

本システムでのPAM は各情報家電機器上のRUM ライブラリで実装される．RUM サーバからの認証開始命令を受け取り，RUM ライブラリは

情報家電機器上の設定ファイルをロードし，認証を開始する．パスワードの場合であればプロンプトをユーザに提示し，入力を促す．また，スマートカードの場合はカードの挿入を待つプログラムを開始する．ユーザからの入力を受け，RUMサーバにデータを送信し，認証を行う．基本的に認証評価はRUMサーバ内で行われる．

4.7　アクセス制御部

本節ではアクセス制御部の設計について述べる．アクセス制御部はRUMサーバの中心的な役割を果たす．通信部からアクセス制御評価要求を受け取り，アクセス制御評価を行う．要求元の機器名，関数名およびユーザ名を入力として受け取る．図4.4にアクセス制御部の構成とRUMライブラリとの関連を示す．

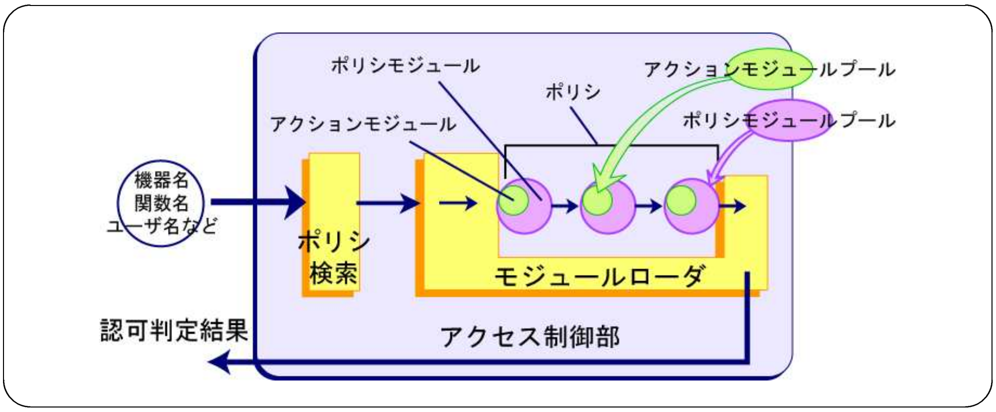


図 4.4: アクセス制御部

4.7.1　ポリシの検索

情報家電機器やその機能に適応的なアクセス制御を行うために，本システムではアクセス制御ポリシの検索を行う．ポリシの検索は情報家電機器情報と関数情報をキーにして行われ，検索結果としてポリシ記述ファイルが返される．表4.2にポリシの検索に利用される項目を示す．

表 4.2: ポリシの検索

	説明
機器情報	機器名や識別子など機器を一意に識別
関数情報	関数名や引数，返り値など
ポリシ記述ファイル	アクセス制御ポリシを記述

4.7.2 ポリシの設計

本システムにおいて，アクセス制御ポリシは複数のポリシモジュールで構成され，ポリシ記述ファイルに記述される．個々のモジュールは単機能のアクセス制御評価を行う．

例えば，図 4.5 に示すようにポリシは複数のモジュールの集合で表現される．この例の場合，利用に伴い同伴者と利用時間に関する評価を記述したポリシを表している．個々の「同伴者」や「利用時間」はポリシモジュールで記述される．「利用時間」や「同伴者名」などの設定はポリシモジュール独自のユーザインタフェースを介して行われ，ポリシ記述ファイルには記述されない．ポリシ記述ファイルでは各サブセットの組み合わせのみを記述する．

こういった複数のモジュールを持つポリシを用いる事で，本システムでは複数のアクセス制御を連続して適用できる．また，ポリシ記述ファイルを編集する事によって柔軟なポリシの記述が容易に行える．ポリシ記述ファイルは情報家電メーカが記述し，配布する．ユーザはこのポリシ記述ファイルと機器や機能との対応付けをユーザインタフェースを通して行う．

また，記述ファイルでは評価終了後の挙動を設定できる．例えば，ACL の評価後利用権限がない旨をユーザに通知するダイアログを表示したり，排他制御評価後，現在起動中の処理に割り込みを掛けるといった処理を設定できる．

これらの挙動をアクションと呼び，プログラムで表現される．ポリシ記述ファイル内では評価後に起動するアクションを記述する．

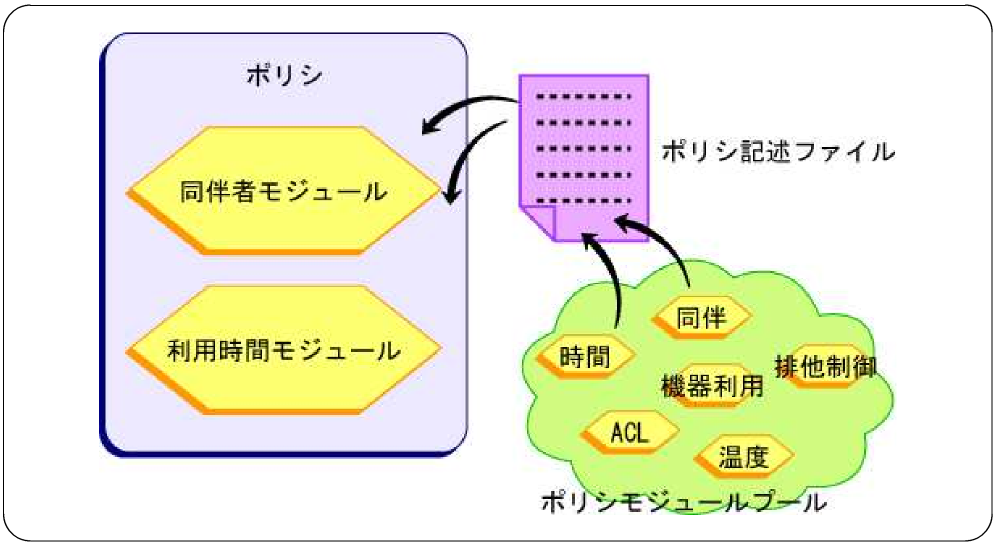


図 4.5: ポリシとポリシモジュール

4.7.3 ポリシの評価

本システムでは，アクセス制御のポリシの評価はポリシ記述ファイルを解釈し，その中に含まれる複数のポリシモジュールをロードすることから始まる．ロードされたポリシモジュールは，個々に評価される．基本的に1つのポリシモジュールはフィルタと環境情報，そしてユーザインタフェースから構成され，評価結果として真か偽のいずれかの値を返す．

フィルタ

取得した環境情報を解釈する閾値や設定．モジュールの評価基準となる．

環境情報

時間情報や位置情報といった環境情報を取得，管理する．モジュールの外に存在する場合もある．

ユーザインタフェース

フィルタの設定確認や変更を行う．

例えば「同伴者がいる時のみ利用可能」といったポリシを表現するポリシモジュールの構成は図 4.6 のようになる．

機能の情報と対象ユーザ，そして同伴者名がフィルタ内に管理される．これにより「機器 A の機能α がユーザ X に利用される際は同伴者 Y を必

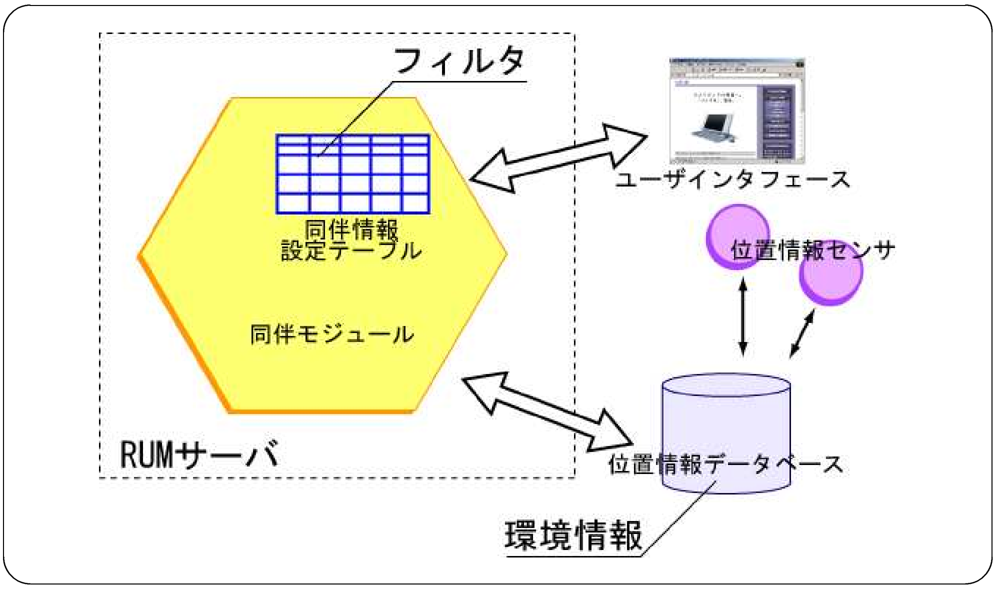


図 4.6: ポリシモジュールの構成

要とする」といった情報が管理される．この情報を基に位置情報データベースを検索し，ユーザ X と同伴者 Y の所在を比較し，検索結果を返す．また，フィルタの設定は独自のユーザインタフェースを通して行われる．

4.8 名前変換部

本システムでは，ユーザは「TV の電源 ON」や「VCR の録画」といった指定で設定の変更や確認を行える事を目指す．しかし，このような表記はシステム側で解釈不能である．システム内では，機器名や関数名を識別子として処理が行われている．従って，システム内でこれらの識別子からユーザが理解可能な形式への変換が必要となる．

4.8.1 機能名の変換

システム内では機能を機器名や関数名を用いて識別している．これらを「VCR の録画」といったユーザが理解可能な形式に変換するため，各情報家電機器ごとに識別子と機能説明との対応付けを行う設定ファイルを用意する．この設定ファイルには機器名や関数名といった実装依存な識別子，機能に関する説明文，ベンダやバージョン情報などを管理する．こ

の設定ファイルを用いて名前変換部が識別子を機能説明文に変換し，ユーザインタフェースに渡す．これにより，ユーザは複雑な機器の内部の情報を意識せずに詳細な設定を容易に行える．

4.8.2 ポリシ名の変換

本システムではポリシに関する詳細な説明はポリシ記述ファイル内に存在する．名前変換部はポリシに関する説明もユーザインタフェースに提供する．従って，ユーザはポリシの変更も，ポリシ名やポリシモジュール名を意識することなく行える．

尚，これらの設定ファイルは情報家電ベンダが用意し配布する．

4.9 本章のまとめ

本章では，まず情報家電適応型アクセス制御モデルを実現するKMSF-RUMシステムの概要について述べた．次に，KMSF-RUMの必要機能をまとめた．そして，システム構成や動作手順について触れた．最後にKMSF-RUMシステムを構成する通信部，認証部，アクセス制御部，名前解決部の詳細設計について述べた．

第5章　KMSF-RUMシステムの実装

本章では，第4章で述べた設計を基にKMSF-RUMシステムの実装について述べる．

KMSF-RUMシステムは，**RUM**サーバや**RUM**ライブラリから構成される．また，RUMサーバ内の通信部や認証部，アクセス制御部，名前変換部などの詳細について触れる．

5.1 実装環境

第4章でも述べたように，KMSF-RUMシステムはRUMサーバと各情報家電上で利用されるRUMライブラリから構成される．

本システムはJava言語を用いてプロトタイプの実装を行った．Java言語を使用することで，マルチプラットホーム性を確保した．コンパイラはJDKバージョン1.3.1を使用した．

各機器はイーサネットとシリアルケーブル，IEEE802.11bのネットワークで接続される．

5.1.1 RUMサーバの実装環境

RUMサーバはホームゲートウェイなどのある程度計算機能力がある機器上での動作を想定している．RUMサーバ内ではリレーショナルデータベースを利用する．また，Webブラウザ経由で各種設定を行うためにWebサーバとサーブレットランタイムを利用した．リレーショナルデータベースとしてPostgreSQL7.0.3を用い，ユーザインタフェース用のサーブレットランタイムとしてJakarta Tomcat 3.2.4を用いた．RUMサーバの実装環境を表5.1に示す．

表 5.1: 実装環境

項目	実装環境
ハードウェア	AT-互換機 K6-III 450MHz ThinkPad X20 Pentium III 700MHz
オペレーティングシステム	Linux 2.4.2 Windows2000 Professional
言語	JDK1.3.1 (J2SE 1.3.1-b24)
リレーショナルデータベース	PostgreSQL 7.0.3
Webサーバ	Apache 1.3.22
サーブレットランタイム	Jakarta Tomcat 3.2.4
認証	JavaCard 2.0

5.1.2　RUM ライブラリの実装環境

RUM ライブラリは情報家電機器上で動作する事を想定している．本システムはプロトタイプの情報家電機器として，慶応義塾大学徳田研究室 Smart Space Lab(SSLab)[20] 内の遠隔制御可能な照明とプラズマディプレイを利用した．現在，これらの機器は直接インターネットに接続出来ないため，外部の RUM ライブラリ動作可能な DUONUS と TINI から RS-232C 経由で制御した．

また，制御用の携帯端末として，iPAQ を利用した．これら使用機器の仕様を表 5.2 に示す．

表 5.2: 情報端末の仕様

項目	**iPAQ**	**DUONUS**	**TINI**
CPU	StrongARM 206MHz	AMD486DX5-133	DS80C390
メインメモリ	64MB	16MB	512K
OS	PocketPC, Linux	FreeBSD 他	TINI OS
外部インタフェース	赤外線 音声入出力他	10Base-T RS-232C	10Base-T RS-232C
その他	TFT 液晶ディスプレイ	駆動部なし	

5.2　通信部の実装

RUM サーバの通信部は RMI[21] を用いて情報家電機器上で動作する RUM ライブラリと通信する．RUM ライブラリは RMI クライアント，RUM サーバは RMI サーバとして動作する．RUM ライブラリからの要求は，RUM サーバの RumRMIIf インタフェース内の check() メソッドが RMI 経由で呼ばれる事によって実行される．RUM ライブラリは check() メソッドに引数として RumInfo オブジェクトを入力し，返り値として boolean 型を得る．getReturnObject() を用いる事で，Rum ライブラリを実装した情報家電機器は，警告やエラーメッセージなどの付加的な情報を RUM サーバから受け取れる．図 5.1 に RumRMIIf インタフェースを示す．

```
package jp.ac.keio.sfc.ht.akihiko.rum.server.rumserver;
import java.rmi.*;
import jp.ac.keio.sfc.ht.akihiko.rum.info.RumInfo;

public interface RumRMIIf extends java.rmi.Remote {
  public boolean check(RumInfo info)
    throws java.rmi.RemoteException;
  public Object getReturnObject()
    throws java.rmi.RemoteException;
}
```

図 5.1: RumRMIIf インタフェース

5.2.1 RumInfo

パッケージやクラス，メソッドなど，情報家電機器上のソフトウェア情報とユーザ情報を基に RUM サーバはアクセス制御を行う．これらのアクセス制御に必要な情報は RumInfo 型オブジェクトに格納され，RUM ライブラリから RUM サーバに渡される．

RumInfo 型のオブジェクトは以下の情報を格納している．

UserInfo UserInfo オブジェクトには，UserID や認証鍵などのユーザに関する情報が格納されている．これらの情報を基にユーザごとにことなるアクセス制御や認証処理が行える．

ResourceInfo パッケージやクラス，メソッドなどアクセス制御を行う範囲の識別に用いる．情報家電機器上のソフトウェアの情報．

EnvInfo ホスト名や IP アドレス，OS 名など情報家電機器プラットフォームの情報を格納している．

sessionID トランザクションの識別子．

通信部では送信されてきた RumInfo を基に認証の必要性を検査する．認証が完了している場合は，アクセス制御部に処理が渡される．図 5.2 に RumInfo インタフェースの詳細を示す．

```
package jp.ac.keio.sfc.ht.akihiko.rum.info;
import java.io.Serializable;

public interface RumInfo extends Serializable{
  public UserInfo     getUserInfo();
  public ResourceInfo getResourceInfo();
  public EnvInfo      getEnvInfo();
  public long         getSessionID();
}
```

図 5.2: RumInfo インタフェース

5.2.2 認証状態管理

通信部では，情報家電機器上の RUM ライブラリからの要求ごとに認証の必要性を管理している．認証が必要な場合には，認証処理の必要性を RUM ライブラリに通知する．すでに認証が行われている場合には，アクセス制御部でアクセス制御の評価が行われる．

認証状態管理は認証状態管理テーブルを用いて行われる．RumInfo オブジェクト内に格納されている情報家電機器ソフトウェアの情報と認証状態管理テーブルで比較が行われる．要求のセッションと一致するエントリが認証状態管理テーブに見つかった場合には，認証処理は省略される．表 5.3 に認証状態管理テーブルの一部を示す．認証状態管理テーブルの各エントリには有効期限があり，有効期限が切れると再度認証が必要となる．

表 5.3: 認証状態管理テーブルの一部

sessionID	**hostname**	**TimeToLive**
893287	133.27.170.100	2002/1/15 09:03:12
203984	133.27.170.101	2002/1/15 10:31:14

5.3 認証部の実装

本節では，認証部の実装について述べる．情報家電機器上のRUMライブラリからの認証要求通信部が受信し，処理を認証部に渡すことで認証処理は開始される．

5.3.1 認証方式の決定

第4章で述べたように，本システムの認証にはPAMを採用した．従って，認証要求によって，情報家電機器上のRUMライブラリから通信部を経て認証部に渡される情報は，RumAuthInfoオブジェクト内で管理される．RumAuthInfoオブジェクト内で管理される情報や以下の3つである．

認証アルゴリズム (algorithm)
クレデンシャルの認証方法を決定する．認証アルゴリズムは，`"MD5"`の様に文字列として，認証部に渡される．

クレデンシャル (credential)
パスワードや識別子などの信用情報．上記の認証アルゴリズムで認証部によって解釈される．

要求 (requirement)
ここのモジュールの全体に与える影響を示す．JAASと同様に，Required，Requisite，SufficientとOptionalの4つの状態を持つ．

認証部では，RUMライブラリから渡されたRumAuthInfoオブジェクトを基に，実際の認証処理を実行する認証モジュールの決定を行う．認証モジュールの決定には，認証モジュール方法決定テーブルが用いられる．認証モジュール方法決定テーブルを，5.4に示す．RumAuthInfoオブジェクトは複数個指定できる．複数指定された場合は，認証方法決定テーブルを用いて認証モジュールを逐次ロードし，実行する．単一のモジュールが全体の認証処理に与える影響については，RumAuthInfoオブジェクト内のrequirement内の値を用いて決定する．

5.3.2 認証処理

実際の認証処理は，ロードされた認証モジュールによって行われる．個々の認証モジュールは，RumAuthIfインタフェースを実装する．ロードさ

表 5.4: 認証方法決定テーブルの一部

Algorithm	**ClassName**
MD5	jp.ac.keio.sfc.ht.akihiko.rum.auth.module.MD5
DES	jp.ac.keio.sfc.ht.akihiko.rum.auth.module.DES

れたモジュールのcheck(RumAuthInfo info)メソッドを呼ぶ事で認証処理が開始される．RumAuthIfインタフェースを図5.3に示す．

```
package jp.ac.keio.sfc.ht.akihiko.rum.auth;

public interface ModuleIf{
  public boolean check(RumAuthInfo info);
}
```

図 5.3: RumAuthIfインタフェース

5.4 アクセス制御部の実装

アクセス制御部はRUMサーバの中心的な役割を果たす．アクセス制御部でのアクセス制御処理の基本的な流れを図5.4に示す．

まず，アクセス制御部ではRumInfoオブジェクト内の情報を用い，適したポリシの検索を行う．次に，検索結果のポリシ記述ファイルを解釈し，必要なモジュールをロードする．さらに，ロードしたモジュールの評価を行う．評価に失敗するかロードするモジュールが無くなるまで，ロードと評価を繰り返す．

最終的にアクションモジュールをロードし，実行を行う．アクセス制御部の処理を終了すると結果は通信部を通じて，RUMライブラリに返される．

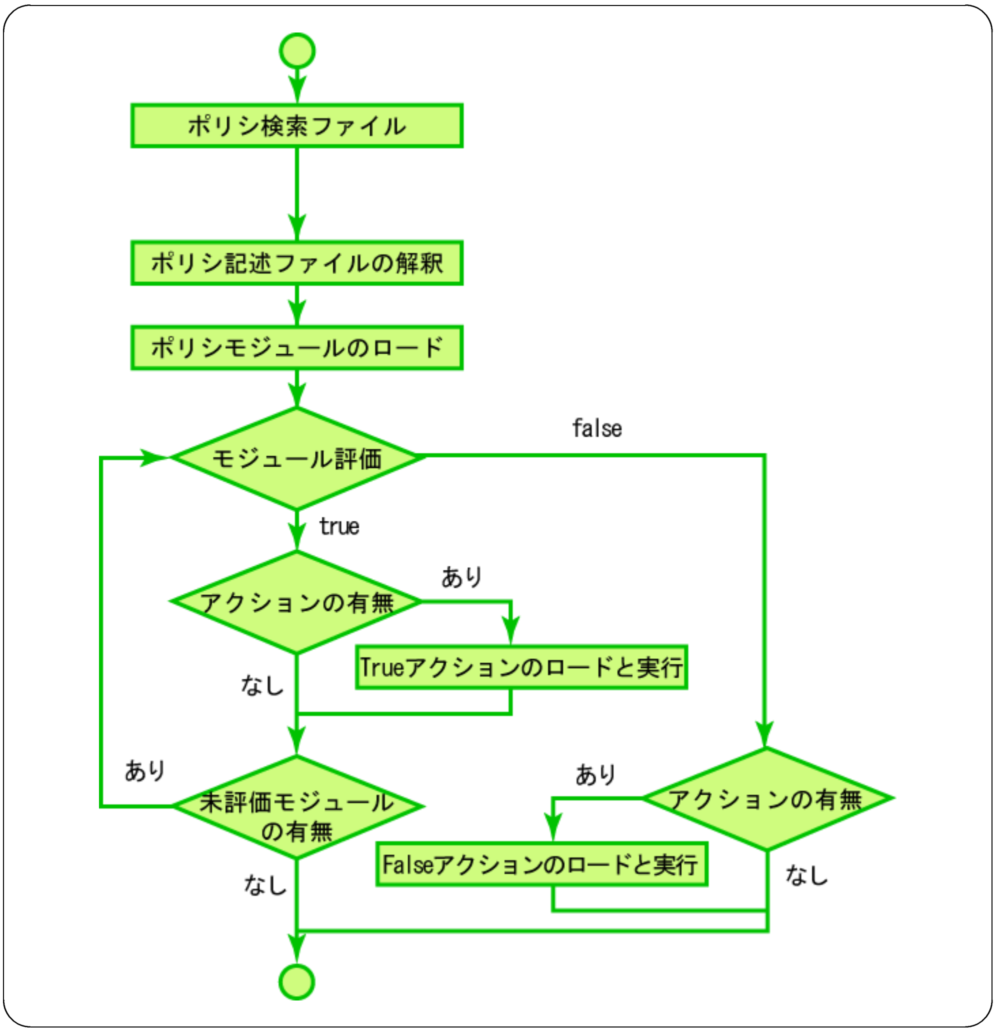


図 5.4: アクセス制御部の処理の流れ

5.4.1　ポリシ検索

アクセス制御部は，情報家電機器上のRUMライブラリから送信されてきたRumInfoオブジェクト内の情報を利用し，適用するアクセス制御ポリシを検索する．検索にはポリシマッピングテーブルを利用する．ポリシの検索は情報家電機器上で動作するソフトウェア資源の情報を基に行われる．

ポリシマッピングテーブルの検索結果として，対応するポリシ記述ファ

イルへのパスが返される．

ポリシマッピングテーブルの各フィールドは以下のとおりである．

resourceID (ID)
情報家電の資源に対して一意な識別子．

resourceGroupID (GID)
情報家電の機能をグループ化する．

hostName (HN)
情報家電ソフトウェアが動作するホスト名．

packageName (PN)
情報家電ソフトウェアのパッケージ名

className (CN)
情報家電ソフトウェアのクラス名

methodName (MN)
情報家電ソフトウェアのメソッド名

arg (ARG)
情報家電ソフトウェアの引数の情報．

returnType (RT)
情報家電ソフトウェアの返り値の情報．

option (OPT)
オプション．

xmlPath (XML)
ポリシを表現するポリシ記述ファイルへのパス．

ポリシマッピングテーブル内のレコードの一例を表 5.5 に示す．

5.4.2 ポリシ記述ファイル

ポリシ検索の結果として返されたパスを基にポリシ記述ファイルを展開する．ポリシ記述ファイル内にはロードするポリシモジュールのクラス名や個々のモジュールの説明，アクションに関する情報が記述される．

表 5.5: ポリシマッピングテーブル

ID	GID	PN	CN	MN	AN	RN	XML
3241	110	package.a	class1				modules/module1.xml
3242	110	package.a	class2				modules/module1.xml
3243	110	package.a	class1	mhd1	null	void	modules/module1.xml
3246	112	org.ak.tv	OFF				modules/module3.xml
3244	111	org.ak.tv	ON	on1	arg1	void	modules/module2.xml
3245	111	org.ak.tv	ON2				modules/module2.xml
3248	113	org.ak.vcr	ON2				modules/module2.xml
3247	113	org.ak.vcr	ON				modules/module2.xml
3249	114	org.ak.vcr	OFF				modules/module2.xml

ポリシ記述ファイルにはポリシモジュールの組み合わせが記述されているため，一つのポリシ記述ファイルから複数のポリシモジュールが生成される．個々のポリシモジュールは単一のアクセス制御評価を実行する．

ポリシ記述ファイルはExtensible Markup Language (XML)[22]で表記され，ModuleLoaderによって解釈される．図5.5にポリシ記述ファイルの一例を，図5.6にポリシ記述ファイルのDocument Type Difinition (DTD)を示す．

アクセス制御部のModuleLoaderはポリシ記述ファイルの記述に従い，逐次ポリシモジュールをロードする．ロードされたポリシモジュールはModuleLoaderによってインスタンス化される．インスタンス化されたポリシモジュールがアクセス制御の評価を個々に行い結果を返す．

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<?xml-stylesheet type="text/xsl" href="module.xsl"?>
<!DOCTYPE modules SYSTEM "modules.dtd">
<modules>
  <name>Companion Module</name>
  <comment>同伴モジュールのサンプル</comment>
  <module>
    <name>Companion</name>
    <comment>指定された人と一緒でなければ利用禁止</comment>
    <class>org.akihiko.module.Companion</class>
    <action>
      <true>
        <name>SampleAction</name>
        <comment>評価が真の時のアクション</comment>
        <class>org.akihiko.action.SampleLog</class>
      </true>
      <false>
        <name>Message</name>
        <comment>評価が偽の時のアクション</comment>
        <class>org.akihiko.action.SampleLog</class>
        <class>org.akihiko.action.SampleMessage</class>
      </false>
    </action>
  </module>
  <module>
    <name>SimpleACL</name>
    <comment>ACL のチェックを行う</comment>
   <class>org.akihiko.module.SimpleACL</class>
  </module>
</modules>
```

図 5.5: ポリシ記述ファイルの例

5.4.3 ポリシモジュール

ポリシモジュールはJavaのクラスとして表現される．ModuleLoaderがポリシモジュールをロードし，インスタンス化する．個々のポリシモジュールはModuleIfインタフェースを実装しなくてはいけない．図5.7にModuleIfインタフェースを示す．ロードされたポリシモジュールはcheck()

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!ELEMENT modules (module+,name,comment)>
<!ELEMENT module  (name,comment,class*,action?)>
<!ELEMENT action  (true,false)>
<!ELEMENT true    (name,comment,class*)>
<!ELEMENT false   (name,comment,class*)>
<!ELEMENT name    (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT class   (#PCDATA)>
```

図 5.6: ポリシ記述ファイルの DTD

メソッドが呼ばれる事で，評価処理を開始する．

```
package jp.ac.keio.sfc.ht.akihiko.rum.server.module;
import jp.ac.keio.sfc.ht.akihiko.rum.info.*;
import jp.ac.keio.sfc.ht.akihiko.rum.lib.RumConf;

public interface ModuleIf{
  public boolean check();
  public void setRumInfo(RumInfo info);
  public void setRumConf(RumConf conf);
}
```

図 5.7: ModuleIf インタフェース

check() メソッドでは RumInfo オブジェクト内のユーザ情報や外部の環境情報サーバなどと基にアクセス制御の評価を行う．時間情報や位置情報，他の機器の情報などポリシモジュールが評価基準にする情報は多種多様である．

したがって，本システムでは check() メソッド内の評価方法はポリシモジュールの実装に依存する．基本的には，環境情報とその環境情報の評価基準となる条件情報の 2 つを基に，ポリシモジュールは評価を行う．

例えば，「UserA と UserB が同室にいる時のみ，UserA が機器α を利用できる」といったポリシを表現する同伴モジュールを考える．個々の UserA や UserB の位置情報は位置情報データベース内で管理され，対象ユーザや対象機器などの条件は同伴条件管理データベースで管理される．これら 2 つのデータベースを用い同伴モジュールは UserA の機器利用の判断

を下す．

モジュールの実装例については第6章で詳しく触れる．

5.4.4 アクションモジュール

本システムでは，各ポリシモジュールの評価後の振る舞いをアクションとして定義する．利用不可の通知をユーザに行う場合や排他制御を行う場合などにアクションを用いる．

ポリシモジュールと同様に，アクションの処理はJavaのクラスとしてアクションモジュールに記述される．アクションモジュールとポリシモジュールの組み合わせはポリシ記述ファイル内の記述される．ポリシモジュールの評価結果が真の場合と偽の場合それぞれに複数ずつアクションを指定できる．ポリシモジュールの評価後，ModuleLoaderが適切なアクションモジュールをロードし，インスタンス化する．その後，アクションモジュール内のinvoke()メソッドが呼ばれる事で，アクションの処理が開始される．各アクションモジュールはActionIfインタフェースを実装しなくてはならない．

ActionIfインタフェースを図5.8に示す．

```
package jp.ac.keio.sfc.ht.akihiko.rum.server.module.action;
import jp.ac.keio.sfc.ht.akihiko.rum.info.*;

public interface ActionIf{
  public void invoke();
  public void setRumInfo(RumInfo info);
}
```

図 5.8: ActionIf インタフェース

5.5 名前変換部の実装

名前変換部はシステム内利用される各種識別子や実装に依存した資源名などをユーザが理解できる機能名や機器名に変換する．ユーザがRUMサーバのマッピングテーブル設定やその他の設定を行う際のユーザインタフェースに，名前変換部の機能は利用される．ユーザはorg.akihiko.tv.ON

といった実装依存の名称は理解できない．名前変換部の機能を利用することで，「TV1 の電源 ON」といった指定でユーザは設定が可能になる．

5.5.1 変換情報

名前解決部は大きく分けて 3 種類の変換を行う．資源名変換，ユーザ名変換そしてポリシモジュールである．

資源名変換
  クラス名，パッケージ名などの実装依存の名称をユーザが理解可能な形式に変換する．変換には情報家電設定ファイルを用いる．

ユーザ名変換
  システム内で利用されるユーザ ID をユーザ名に変換する．変換には認証部内のユーザ情報管理テーブルを利用する．

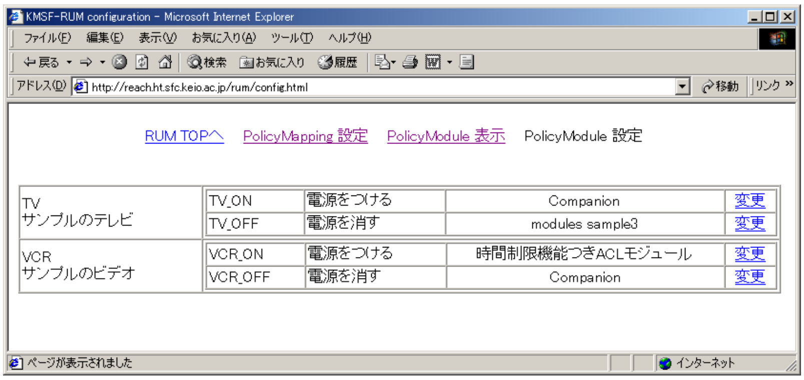
ポリシモジュール名変換
  システム内で利用されるポリシ名やポリシモジュール名をユーザが理解可能な形式に変換する．変換にはポリシ記述ファイルが利用される．

5.5.2 ユーザインタフェース

名前変換部が機能を提供するユーザインタフェースは大きく分けて 2 種類ある．RUM サーバ設定用のユーザインタフェースとポリシモジュール内の各種設定を行うユーザインタフェースである．

RUM サーバ設定ユーザインタフェース
  ポリシマッピングテーブルの設定や認証部内のパスワード変更など RUM サーバの設定を行うユーザインタフェース．RUM サーバ設定ユーザインタフェースでは，ユーザ名変換や資源名変換，ポリシモジュール名変換が利用される．ポリシマッピングテーブルの設定を行うポリシ設定ユーザインタフェースを図 5.9 に示す．

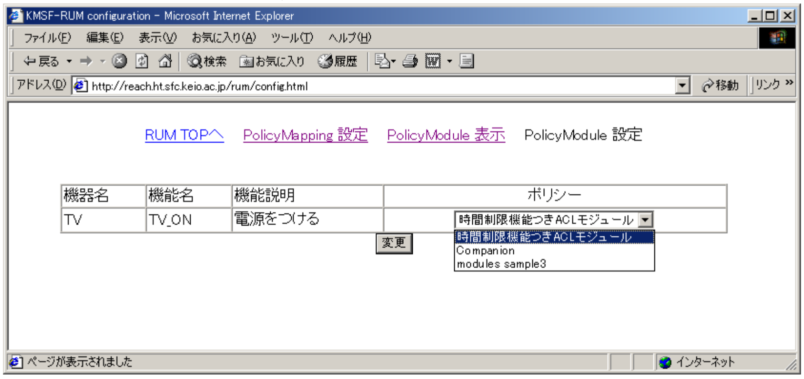
ポリシモジュール設定ユーザインタフェース
  ポリシモジュール内では環境情報の評価基準を設定する必要がある．例えば，「ユーザ A は午前 10 時から午後 5 時まで利用禁止」といった



RUM TOPへ PolicyMapping 設定 PolicyModule 表示 PolicyModule 設定

TV サンプルのテレビ	TV_ON	電源をつける	Companion	変更
	TV_OFF	電源を消す	modules sample3	変更
VCR サンプルのビデオ	VCR_ON	電源をつける	時間制限機能つきACLモジュール	変更
	VCR_OFF	電源を消す	Companion	変更

変更対象機能選択画面



RUM TOPへ PolicyMapping 設定 PolicyModule 表示 PolicyModule 設定

機器名	機能名	機能説明	ポリシー
TV	TV_ON	電源をつける	時間制限機能つきACLモジュール

時間制限機能つきACLモジュール
Companion
modules sample3

ポリシ変更画面

図 5.9: ポリシ設定ユーザインタフェース

> 時間限定のポリシモジュールを考える．この場合，時間や対象ユーザの設定を変更するためのユーザインタフェースが必要となる．これらのポリシモジュール設定ユーザインタフェースに対して，名前変換部はユーザ名変換やポリシモジュール変換を提供する．

各種ユーザインタフェースに対して，名前の変換機能を提供する事で，ユーザは実装に依存した名称や識別子を意識せずに様々な設定が可能となる．

5.5.3　情報家電定義ファイル

環境情報定義ファイルは資源名変換に利用される．クラス名やパッケージ名などの実装に依存した名称とユーザが理解可能な機能名や機能説明

との関連付けを行う．情報家電定義ファイルは情報家電開発ベンダが情報家電機器と共に出荷する．情報家電定義ファイルはXMLで記述される．図5.10に情報家電記述ファイルの例，図5.11に情報家電記述ファイルのDTDを示す．DTDでは，文書の構造を表現するための要素の定義を行う．

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!DOCTYPE appliance SYSTEM "appliance.dtd">
<appliance>
  <name>TV</name>
  <info>
    <vender>vender</vender>
    <version>v1.0</version>
    <comment>サンプルのテレビ</comment>
  </info>
  <function>
    <name>TV_ON</name>
    <comment>電源を入れる</comment>
    <software>
      <package>org.akihiko.tv</package>
      <class>ON</class>
      <method>method_on</method>
      <arg>null</arg>
      <returnType>void</returnType>
    </software>
    <software>
      <package>org.akihiko.tv</package>
      <class>ON2</class>
      <method></method>
    </software>
  </function>
  <function>
    <name>TV_OFF</name>
    <comment>電源を切る</comment>
    <software>
      <package>org.akihiko.tv</package>
      <class>OFF</class>
      <method></method>
    </software>
          ····以下省略 ····
```

図 5.10: 情報家電記述ファイルの例

要素`<function>`は「電源を入れる」などのユーザが理解可能な情報家電の機能の最小単位を表す．情報家電機器は複数の機能によって構成され

るため，情報家電記述ファイル内にも複数の要素`<function>`が存在する．

情報家電機器上の実装に依存したクラス名やパッケージ名などの資源の情報は要素`<software>`で表現される．要素`<function>`内には複数の要素`<software>`を持つことが出来る．つまり，一つの情報家電機器の機能に複数の資源を対応付けることが可能である．これにより，異なるパッケージやクラスがTVの電源を入れるという同一の機能を提供可能である．

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!ELEMENT appliance  (name,info,env?,function*)>
<!ELEMENT info       (vender,version,comment)>
<!ELEMENT env        (hostname,VMName,MWName)>
<!ELEMENT function   (name,comment,software*)>
<!ELEMENT software   (package,class,method,arg?,returnType?)>
<!ELEMENT name       (#PCDATA)>
<!ELEMENT comment    (#PCDATA)>
<!ELEMENT package    (#PCDATA)>
<!ELEMENT class      (#PCDATA)>
<!ELEMENT method     (#PCDATA)>
<!ELEMENT arg        (#PCDATA)>
<!ELEMENT returnType (#PCDATA)>
<!ELEMENT hostname   (#PCDATA)>
<!ELEMENT VMName     (#PCDATA)>
<!ELEMENT MWName     (#PCDATA)>
<!ELEMENT vender     (#PCDATA)>
<!ELEMENT version    (#PCDATA)>
```

図 5.11: 情報家電記述ファイルのDTD

5.6 RUMライブラリの実装

RUMライブラリは情報家電機器上のアプリケーションで利用される．主な目的は認証要求とアクセス制御評価要求をRUMサーバに送信し，結果を得る事である．RUMライブラリの機能はjp.ac.keio.sfc.ht.akihiko.rum.libパッケージ内で提供される．

5.6.1 認証と識別

本システムは第4章で述べた様に，PAMモデルを採用している．このため，RUMライブラリでは，まず設定ファイルに記述された認証モジュー

ルをロードする．次に，ロードされた認証モジュールが，ユーザから認証のための入力を受け付ける．ユーザからの入力と認証方式をRUMサーバに送信する．実際の認証処理はRUMサーバの認証部で行われる．

認証モジュール

認証方式は情報家電機器によって異なる．例えば，指紋認証が適している情報家電機器や文字入力によるパスワード入力を必要とする情報家電機器など様々である．また，RSAやDESなど認証アルゴリズムに関しても複数のアルゴリズムが存在する．

本システムでは，これら認証方法と認証アルゴリズムを認証モジュール内に隠蔽し，認証モジュールを組替える事で，認証方法と認証アルゴリズムの変更を実現する．同時に複数の認証モジュールをロードすることで，連続して異なる方法で認証を行う事が出来る．これにより，スマートカードで認証した後，パスワード入力による認証を行う事も可能になる．

認証モジュールの指定や組み合わせは情報家電機器上の認証設定ファイル内に記述される．認証モジュールはJava言語のクラスとして表現される．

RUMライブラリの認証処理が開始されると，まず認証設定ファイル内に指定されているモジュールがロードされ，インスタンス化が行われる．その後，ユーザにパスワードの入力やカードの挿入などを促し，入力待ちに入る．ユーザの入力を受け，RUMライブラリは入力された鍵と選択された認証アルゴリズムをRUMサーバに送信する．認証が成功した場合，トランザクションの識別子をRUMサーバから受け取る．

ログインとログアウト

認証が成功するとトランザクションの認証状態がRUMサーバ内で一定時間保持される．有効期限内は認証済みトランザクションは再度認証を必要とすることなく，アクセス制御要求を実行できる．

機器や機能の使用が終了した際など，明示的にRumClientクラスのlogout()メソッドを呼ぶことでRUMドメインからのログアウトが可能となる．

5.6.2 アクセス制御要求

認証後，RUM ライブラリは RumClient クラスの check(RumInfo info) メソッドを呼ぶ事で RUM サーバに対してアクセス制御評価要求を送信する．check(RumInfo info) メソッドでの評価が成功した場合はクリティカルセクションの実行を行う．評価が失敗した場合，RumAccessControlException を発行する．

5.6.3 利用例

図 5.12 に実際の RUM ライブラリの利用例を示す．以下の部分を情報家電機器のソフトウェアに追加する事で，環境情報適応型アクセス制御を実現できる．

まず始めに client.login() メソッドによって認証処理を行う．次に，RumUtil.packInfo() メソッドで RUM サーバに送信する資源情報を設定する．そして，client.check(Info info) メソッドを呼ぶ事で，アクセス制御の評価を RUM サーバに依頼する．評価が失敗した時の情報などは RumAccessControlException を通じて取得する．

```
RumInfo info = null;
RumClient client = new RumClient();
try{
   /*** 認証処理 ***/
  client.login();
   /*** 資源情報を設定 ***/
  info = RumUtil.packInfo(this,"foo");
   /*** アクセス制御評価 ***/
  client.check(info);
   /*** 以下にクリティカルセクションを記述 ***/
  foo();
}catch(RumLoginException rle){
   /*** 認証失敗の処理 ***/
}catch(RumAccessControlException rac){
   /*** 評価結果がアクセス禁止の場合の処理 ***/
   String message = rac.getMessage();
}
```

図 5.12: 情報家電機器上での利用例

第6章　アプリケーション

本章では，RUMシステムを用いたアプリケーションについて述べる．

アプリケーションは，情報家電機器を制御する情報家電ソフトウェアとアクセス制御を行うポリシモジュールから構成される．

6.1　情報家電アプリケーション

KMSF-RUM システムを利用した情報家電アプリケーションとポリスモジュールのプロトタイプを実装した．

6.1.1　システム構成

KMSF-RUM システムを利用したアプリケーションは，情報家電機器上に実装された情報家電アプリケーションと RUM サーバ内で動作するポリシモジュールから構成される．また，情報家電アプリケーションは PDA などを用いて遠隔利用が可能である．図 6.1 に KMSF-RUM を利用したアプリケーションのシステム構成図を示す．

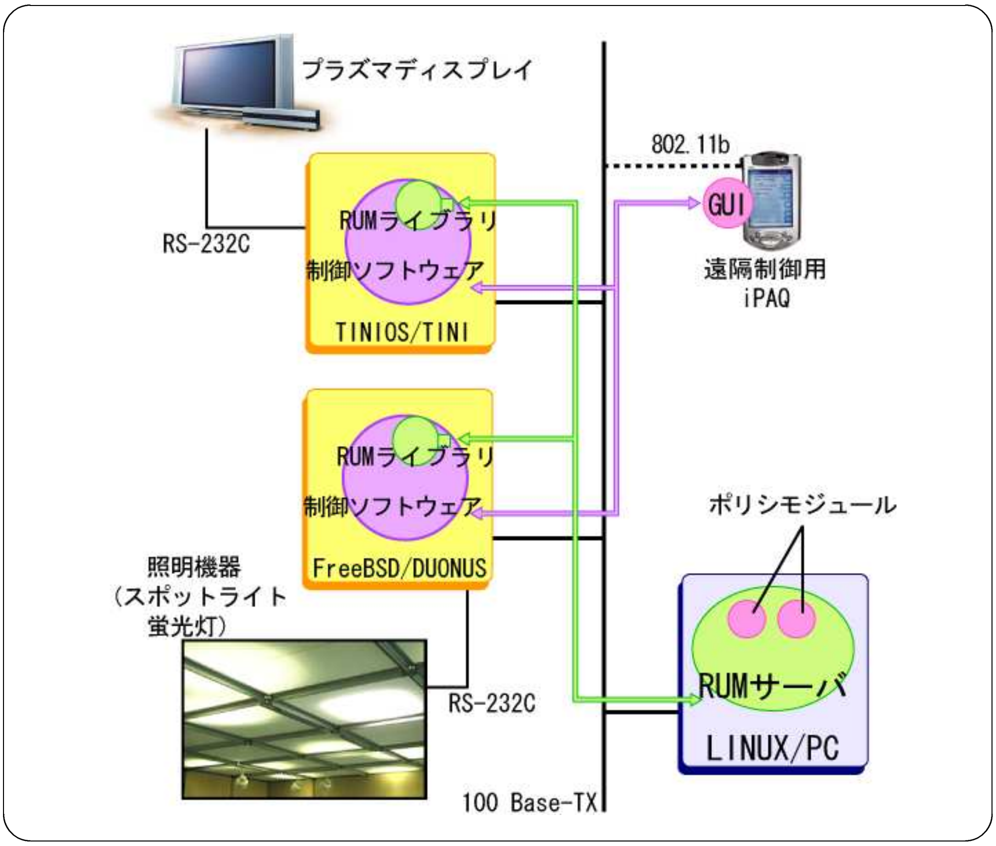


図 6.1: アプリケーションと RUM システム

6.1.2　シナリオ

情報家電機器として調光照明器具とプラズマディスプレイパネル(PDP)を利用して，情報家電アプリケーションを構築した．また，ポリシモジュールとして利用時間制限モジュールと同伴モジュールを実装した．

これらの機器とモジュールを利用し，以下のようなシナリオを実現するアプリケーシュンを構築した．

- ユーザAはユーザBが同室にいなければ，照明機器を制御できない．
- ユーザCは午後23時から午前5時までPDPを利用できない．

また，上記の2つのシナリオを組み合わせ，位置情報と時間情報の両方による利用権の評価を行う事も可能である．

6.2　情報家電アプリケーション

情報家電アプリケーションとして，調光照明器具とPDPを制御するアプリケーションを構築した．現在，CやJavaなどの高級言語によってプログラム可能な家電は少ない．利用した調光照明器具とPDPも機器内にアプリケーションを実装する事はできない．従って，TINIとDUONUSなどの小型計算機上にアプリケーションを実装し，シリアルケーブル経由で間接的に制御した．

6.2.1　調光照明器具

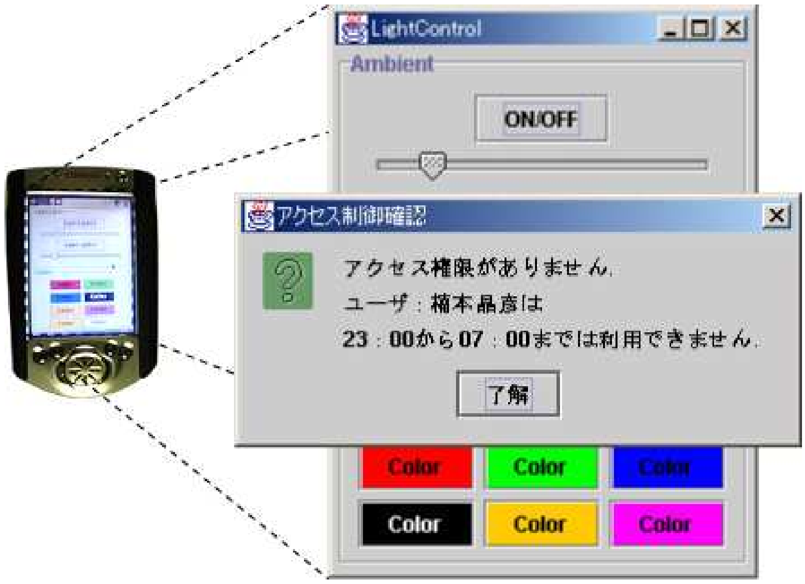
照明制御アプリケーションは，DUONUS上のFreeBSDにインストールされたJavaVM上で動作し，シリアルケーブルで接続された照明制御盤を間接的に制御する．具体的には，照明制御アプリケーションは慶応義塾大学徳田研究室SSLab内の天井部の5個の蛍光灯と5個のスポットライトのオン/オフを制御する．

照明制御アプリケーションは，Compaq Computer社のPDA(Personal Digital Assistants)であるiPAQ Pocket PC上のGUI(Graphical User Interface)から制御される．照明制御アプリケーションは，GUIからの制御命令を受け，認可要求をRUMサーバに送る．RUMサーバは，ポリシマッピングテーブルを基づき照明制御アプリケーションに対するアクセス制御ポリシを決定し，決定されたアクセス制御ポリシを評価する．評価結

果は照明制御アプリケーションに返送され，利用が認可された場合には処理を継続し，認可されない場合は結果を PDA 上の GUI を通じてユーザに提示する．

6.2.2　プラズマディプレイパネル

PDP 制御アプリケーションは，TINI 内の TINIOS 上で動作する．PDP 制御アプリケーションも照明制御アプリケーションと同様に，PDA 上の GUI からの要求を受け，シリアルケーブルを用いて間接的に PDP を制御する．また，PDP 利用の認可は，RUM サーバを用いて評価される．図 6.2 に PDA 上の GUI を示す．



アクセス権限がありません．
ユーザ：楠本晶彦は
23：00から07：00までは利用できません．

図 6.2: PDA 上の制御用 GUI

6.3　ポリシモジュール

KMSF-RUM システムを用いたアプリケーションの一部として，ポリシモジュールの実装を行った．実装したポリシモジュールは，利用時間制限モジュールと同伴モジュールである．

6.3.1 利用時間制限モジュール

利用時間制限モジュールは，特定機能の利用を時間帯に応じて制限する事を目的とする．例えば，午前1時から午前6時まではTVの利用を制限することが考えられる．

SimpleTimerクラスとTimerテーブル，ユーザインタフェースで構成される．

SimpleTimerクラス

ModuleIfインタフェースを実装したポリシモジュール．RUMサーバにロードされ，実際の評価処理を行う．クラス内部で現在の時刻を管理し，Timerテーブルの情報と比較する．第4.7.2項で述べた分類の環境情報は，SimpleTimerクラス内で管理される現在時刻に対応する．

Timerテーブル

リレーショナルデータベースのテーブルとして実装される．内部で管理している主な情報を以下に示す．前述分類では，フィルタに相当する．

- 対象機能（PackageName,ClassName,etc）
- 対象ユーザ(UserName)
- 利用制限開始時間(Begin)
- 利用制限終了時間(End)

ユーザインタフェース

Timerテーブル内の設定情報の確認や変更に利用される．Java Servletとして実装され，Webブラウザ経由で用いられる．また，RUMシステム内の名前変換部を利用し，ユーザ名や機能名の変換が行われる．したがって，ユーザは「プラズマディスプレイの電源を入れる」といった指定で設定ができる．

6.3.2 同伴モジュール

同伴モジュールは，特定機能を対象ユーザが単独で利用することを制限する．このモジュールを利用することで，「ユーザα が機器Aの機能1を利用する際には，同室にユーザβ が存在しなくてはいけない」といっ

たポリシの指定が可能となる．同伴モジュールは，子供などが親の目の届かないところで，調理器具などの危険な機器の利用を制限できる．

同伴モジュールは，Companionクラスと Companionテーブル，位置情報データベース，ユーザインタフェースにより構成される．

Companion クラス

ModuleIf インタフェースを実装したポリシモジュールで，RUM サーバにロードされ，実際にアクセス制御の評価処理を行う．Companion クラスは，Companion テーブルの設定情報に基づき，位置情報データベースを検索し，アクセス制御の評価を行う．

Companion テーブル

同伴モジュールの設定情報が管理される．Companion テーブルが管理する情報を以下に示す．前述の親子の例を用いると，対象ユーザと同伴指定ユーザは，前者が子供で後者が親に対応する．また，範囲は「部屋」，「フロア」や「建物」などを指定し，同一空間として認定する粒度を指定する．フィルタとして動作する．

- 対象機能（PackageName,ClassName,etc）
- 対象ユーザ (UserName)
- 同伴指定ユーザ (TargetName)
- 範囲 (Area)

位置情報データベース

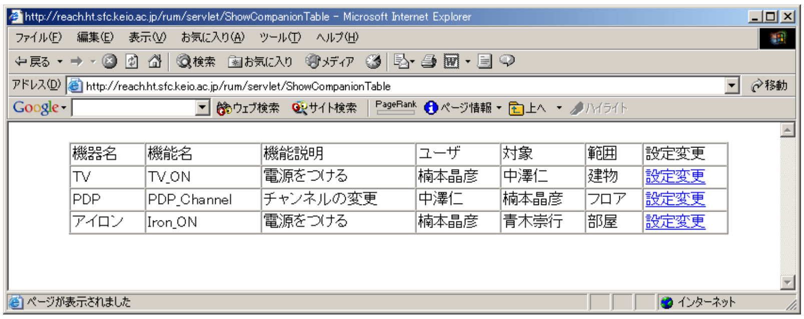
位置情報データベースは，RF-ID[23] を利用し，ユーザの位置情報を管理する．RF-ID のタグの情報をリーダが感知し，位置情報データベースに登録する．位置情報はデータベース内で表 6.1 の様な形式で管理される．

表 6.1: 認証状態管理テーブルの一部

UserName	**TagID**	**Location**
akihiko	023	living.2ndfloar
soko	037	bathroom.1ndfloar
jin	079	out

ユーザインタフェース

ユーザインタフェースは，Java Servlet として実装され，Companion テーブルの設定情報の確認や変更を行う．利用時間制限モジュール同様，RUM システム内の名前変換部を利用しているため，容易な確認や設定が可能である．図 6.3 に，同伴モジュールの設定確認画面を示す．



機器名	機能名	機能説明	ユーザ	対象	範囲	設定変更
TV	TV_ON	電源をつける	楠本晶彦	中澤仁	建物	設定変更
PDP	PDP_Channel	チャンネルの変更	中澤仁	楠本晶彦	フロア	設定変更
アイロン	Iron_ON	電源をつける	楠本晶彦	青木崇行	部屋	設定変更

ページが表示されました

図 6.3: 同伴モジュール設定確認画面

6.3.3　モジュールの複合利用

利用時間制限モジュールと同伴モジュールの 2 つのモジュールに加え，同時使用機器指定モジュールも実装した．そして，これら 3 つのポリシモジュールの組み合わせを記述したポリシ記述ファイルを作成した．これにより，位置情報と時間情報，さらには他の機器の動作状況を考慮したアクセス制御を実現した．これにより，「ユーザ A が機器α の機能 1 を利用する際には，午前中にユーザ B 同伴で，機器β の機能 2 が動作して無くてはいけない．」といったアクセス制御が可能となる．

6.4　本章のまとめ

本章では，RUM システムを用いたアプリケーションについて述べた．

アプリケーションは，情報家電機器を制御する情報家電ソフトウェアとアクセス制御を行うポリシモジュールから構成される．

情報家電ソフトウェアとして，プラズマディスプレイパネルと調光照明機器を制御する，照明制御アプリケーションと PDP 制御アプリケーショ

ンを実装した．これらの情報家電ソフトウェアは，PDA 上の GUI から制御し，RUM システムを利用したアクセス制御が適用される．

アクセス制御を行うポリシモジュールとして，利用時間制限モジュール，同伴モジュールと同時使用機器指定モジュールを実装した．これらポリシモジュールを用いる事で，時間情報や位置情報，他の機器の稼動情報などを利用したアクセス制御を実現した．

また，ポリシ記述ファイルを作成する事で，単一のポリシモジュールを組み合わせ，複雑な状況指定を実現した．

第7章　評価と考察

本章では，KMSF-RUM システムの評価を行う．

まず，本システムの基本性能評価を中心とした，定量的評価を行う．その後，KMSF-RUM システムの定性的評価を行い，本システムの有用性について述べる．

7.1 定量的評価

KMSF-RUMシステムが実用に耐えうる事を証明する為，本研究では定量的評価を行った．

7.1.1 測定環境

定量的評価のための測定は，表7.1に示す環境で行った．

表 7.1: 測定環境

項目	実装環境
ハードウェア	AT-互換機
CPU	AMD K6-III 450MHz
メモリ	128MB
オペレーティングシステム	Linux 2.4.2
言語	JDK1.3.1 (J2SE 1.3.1-b24)
リレーショナルデータベース	PostgreSQL 7.0.3
利用アルゴリズム	MD5

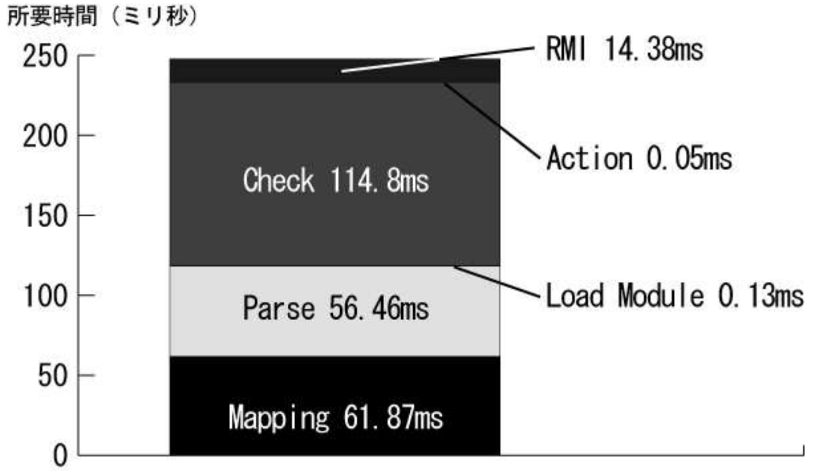
本システムは家庭内ネットワーク内での利用を想定する．従って，RUMライブラリが動作する情報家電機器上とRUMサーバの間の通信によるオーバヘッドは，極めて小さいと考えられる．そのため今回の測定は，RUMライブラリとRUMサーバが同一ホスト内に存在する環境で行った．

7.1.2 測定結果

前述の測定環境で基本性能評価を行った．行った評価は，RUMシステム内の所要時間の評価，機能数とポリシモジュール数がシステムの実行速度に与える影響の評価である．

RUMシステム内の所要時間

RUMシステム内の所要時間の分布を測定し，評価を行った．測定結果を図7.1に示す．



Mapping :アクセス制御ポリシ検索に要する時間
Parse :ポリシ記述ファイル展開に要する時間
LoadModule :ポリシモジュールのロードに要する時間
Check :ポリシモジュールの評価時間
Action :アクションモジュールのロードと起動に要する時間
RMI :RMI のオーバヘッド

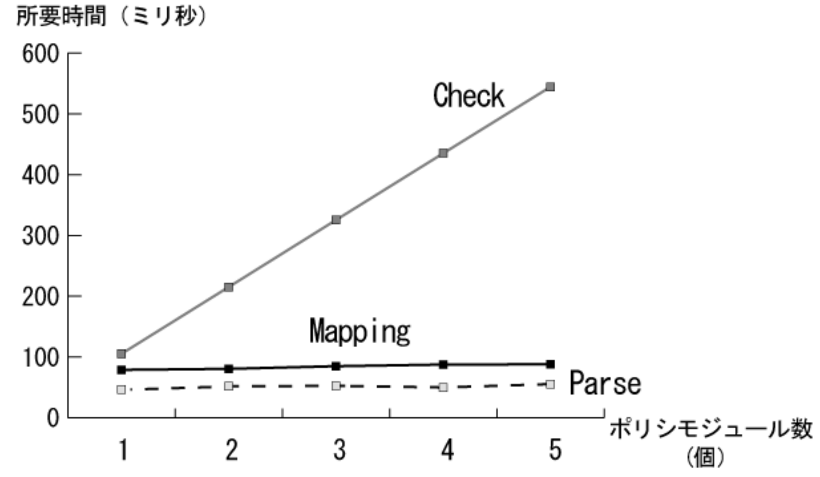
図 7.1: RUM システム内の所要時間の分布

測定には，第 6 章で述べた同伴モジュールを利用した．また，RUM サーバ内では 100 個の情報家電の機能を管理し，位置情報データベース内では 10 名の位置情報を管理している．応答時間が 250 ミリ秒以内であり，情報家電での機器制御の際に行われるアクセス制御である事を考慮すれば，十分に実用に耐えうるといえる．

ポリシモジュールのオーバヘッド

ポリシモジュール数の増加が，システム全体の実行速度に与える影響を測定，評価した．評価に 100 ミリ秒要するポリシモジュールの個数を変化させ，所要時間を計測した結果を図 7.2 に示す．

複数のポリシモジュールを連結して評価する場合，評価に要する時間はポリシモジュールの個数に比例する事が分かる．現状では，ポリシモ



Mapping　:アクセス制御ポリシ検索に要する時間
Parse　:ポリシ記述ファイル展開に要する時間
Check　:ポリシモジュールの評価時間

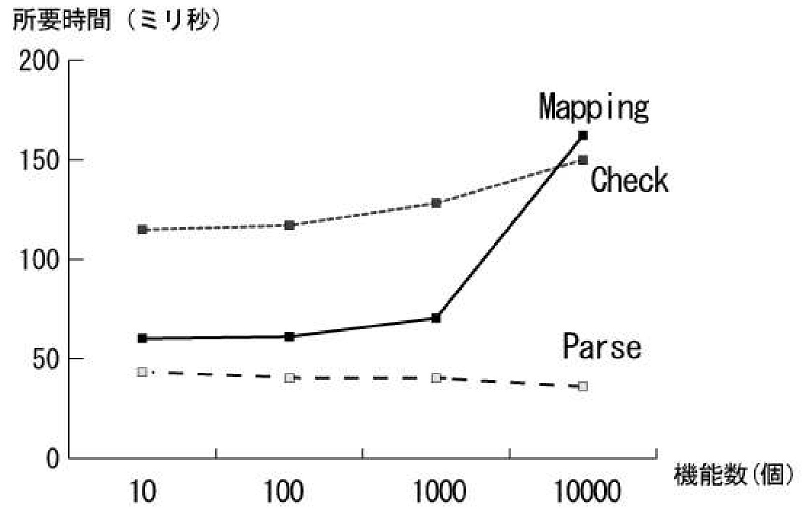
図 7.2: ポリシモジュールのオーバヘッド

ジュールの評価に要する時間は，ポリシモジュールの実装依存である．

ポリシマッピングテーブルのオーバヘッド

ポリシマッピングテーブルにおいて，機能数がRUMシステムの実行速度に与える影響を測定し，評価を行った．測定方法は，RUMサーバ内で情報家電の機能と機能に対応するポリシ記述ファイルを管理するポリシマッピングテーブルのレコード数を変化させた．レコード数は情報家電の機能数に対応する．機能数を10個から10000個まで，増加させた時の所要時間を測定した結果を，図7.3に示す．

機能数の増加に伴い，予想通りポリシマッピングテーブルのオーバヘッドが増加した．機能数1000個では70ミリ秒以内に抑えられたが，10000個の場合には170ミリ秒の所要時間を要した．しかし，情報家電機器の機能数の総和は，家庭内ネットワーク内で利用されることを考慮すれば，10000個程度十分であり，170ミリ秒は許容範囲であると言える．



Mapping :アクセス制御ポリシ検索に要する時間

Parse :ポリシ記述ファイル展開に要する時間

Check :ポリシモジュールの評価時間

図 7.3: ポリシマッピングテーブルのオーバヘッド

7.2 定性的評価

本論文では，KMSF-RUM システムの有効性を示すため定性的評価を行った．

7.2.1 直感的なアクセス制御ポリシの記述

KMSF-RUM システムでは，環境情報を利用したアクセス制御ポリシの記述や指定が可能である．これにより，ユーザは「時間情報」や「位置情報」，「他の機器の稼動状況」などを利用したアクセス制御の指定や変更ができる．

従来の計算機環境でのアクセス制御は，対象となる計算機資源にアクセスするユーザ名やそのロール，権限などに基づいて行われてきた．従来の方法によるアクセス制御ポリシの設定や記述は，ビジネスの場での利用を想定し，開発されたものが中心である．従って，機器利用に関する

知識の乏しいユーザの利用が想定される情報家電環境において，複雑なユーザ名やそのロール，権限に関する管理や設定を強いることは難しい．

このような環境下では，「深夜の利用禁止」や「親が同伴であれば利用可」といった日常的に使用している状況によるアクセス制御ポリシの記述や設定が重要である．

7.2.2 ヘテロジニアスな環境への適応

情報家電環境は，パーソナルコンピュータやワークステーションなどの既存の計算機環境と比較し，ヘテロジニアスな環境であると言える．多種多様な機器上で様々なソフトウェアが，混在するプロトコルにより通信を行うからである．また，幅広いユーザ層によって利用される点においても，既存の計算機環境と異なる．

従来のアクセス制御は，オペレーティングシステム内，単一の機器内やシステム内で行われてきた．従って，ヘテロジニアスな環境に対する適応は困難であった．本システムでは，Java 言語を用い実装した RUM ライブラリがミドルウェアとして動作し，RUM サーバと通信しながらアクセス制御を実現する．従って，マルチプラットフォームでの動作を実現し，Jini や HAVi などの異なるミドルウェア上のサービスからの利用も可能である．

閉じたシステム内でのアクセス制御に比べ，本システムのアプローチは高信頼なアクセス制御の実現は困難である．しかし，ホームゲートウェイで隔離された家庭内ネットワークでは，外部の悪意ある意図的な攻撃より，ユーザの経験不足に伴う誤操作や故障からのシステムやユーザの保護が重要である．外部の悪意ある意図的な攻撃への対策は，ホームゲートウェイなどのインターネットと家庭内ネットワークの境界上でファイアウォールなどの技術が研究されている．

7.2.3 ユーザビリティ

従来の計算機システムでは，特別な知識と能力を持つ管理者がアクセス制御のポリシの変更やポリシ内の条件設定などを行っていた．また，知識をもった管理者でさえ，ACL の適切な適用や管理，運営は困難であった．情報家電環境においては，ユーザ自身が特別な知識を必要とせずに，アクセス制御を行える事が重要である．

本システムでは，名前変換部の利用することで，簡便なユーザインタフェース提供できる．ユーザインタフェースは，アクセス制御ポリシと情報感電機器内の機能との対応付けや個々のポリシモジュール内の条件設定などに利用される．ユーザはこれらのユーザインタフェースを，実装に依存した名称を意識せずに利用可能である．

また，上記の直感的なアクセス制御ポリシの記述や指定が可能な点も，既存のアクセス制御を実現するシステムと比較し，ユーザビリティが向上した点である．

7.2.4 簡便な実装

本システムでは RUM サーバを提供すると同時に，情報家電機器上で RUM サーバの機能を利用したり，ポリシモジュールを容易に実装するためのライブラリ群を提供する．これらのライブラリを利用することで，容易に環境情報適応型アクセス制御モデルを実現する情報家電システムを構築できる．

第 6 章で実装したポリシモジュールに要したコードの行数は以下の通りである．

- SimpleTimer クラス—-111 行
- Companion クラス——114 行

実際のアクセス制御をプログラマが記述する部分は 80 行前後である．

また，ポリシ記述ファイルによって，既存のポリシモジュールの組み合わせを行う事で，プログラマは新たなポリシモジュールのコードを実装することなく複雑なポリシ記述を実現できる．

7.3 本章のまとめ

本章では定量的，定性的両評価を行った．

定量的評価では，システム内の処理速度の内訳，機能数やポリシモジュール数がシステムの実行速度に与える影響などを評価し，本システムが実用に耐えうるものであることを実証した．

また，定性的評価では，直感的なアクセス制御ポリシの記述，ヘテロジニアスな環境への適応，ユーザビリティや簡便な実装などの点における既存システムと比較した優位性について述べた．

定量的，定性的両評価を通じて，本システムの有効性を示した．

第8章　終わりに

8.1　今後の課題

本章では，現在RUMシステムに残されている課題について触れる．

8.1.1　アクションに関する課題

本研究において，アクションと呼ばれる認可判定後の振る舞いを記述する機構を導入した．現在，アクションに関する実装が不十分であり，アクションが提供する機能はプログラマ依存である部分が大きい．

今後，アクションで提供する機能を抽象化し，共通で利用できるアクションのライブラリ化が必要となる．例えば，調理器を利用する際に，換気扇の動作を．要求するアクセス制御ポリシを考える．現状のRUMシステムでは，換気扇が動作していない場合，調理器の利用要求が棄却され，その旨をユーザに提示する．しかし，アクションによって機器の起動命令を実行する事で，自動的に換気扇が動作するアプリケーションを構築できる．

8.1.2　簡易インストール

現在のRUMシステムでは，RUMサーバに，新規のモジュールやポリシ記述ファイルのインストールする際には，RUMサーバ内に手動でファイル類をコピーする必要がある．

RUMサーバを実用的なシステムにするためには，より簡便なモジュールやポリシ記述ファイルのインストール方法が必要である．今後，RUMシステムへのファイルのインストールの自動化やネットワークを通じたモジュールのダイナミックなロードを考慮する必要がある．

8.1.3 RUMライブラリに関する課題

現在，情報家電ソフトウェア内でRUMライブラリを利用する場合，個々サービスに処理をRUMライブラリを利用した処理を記述しなくてはいけない．Jiniサービスの様に，既存のミドルウェアの上で動作するサービスの場合，プログラムの変更が必要とされる．今後，RUMライブラリを用いた，アクセス制御機能の記述をミドルウェア内で可能にする事で，ミドルウェアを修正するだけで，それらミドルウェア上で動作するサービスを変更することなく，アクセス制御機能を付加できる．

8.2 まとめ

本論文では，まず情報家電環境の特徴について述べ，情報家電環境におけるセキュリティについて論じた．そして，情報家電環境におけるセキュリティモデルとして，環境情報適応型アクセス制御モデルを提案した．また，また，環境情報適応型アクセス制御モデルを実現するKMSF-RUMシステムの設計と実装，評価を行い，本システムの有効性について述べた．

KMSF-RUMシステムは，情報家電環境におけるユーザの誤操作や機器の故障から，ユーザや情報家電システムを保護する事を目的とする．従来のセキュリティ機構は外部から不正アクセスや機密情報の漏洩などといった不正行為からの保護を主な目的としていた．しかし，事故により物理的な損失を被り，主に家庭内ネットワークを通じて利用される情報家電機器は，ユーザの誤操作や機器の故障に対する保護が重要である．そして，ユーザの誤操作や機器の故障に対する取り組みは，従来のセキュリティ機構では不十分である．また，情報家電環境のセキュリティ機構において，ユーザビリティを下げない事は重要である．

そこで，本システムでは時間情報や位置情報，機器の稼動情報などの環境情報を用いポリシを記述し，アクセス制御を行う．これにより，ユーザは「母親と一緒であれば，アイロンの使用を許可する」といった状況によるアクセス制御の指定が可能となる．アクセス制御の指定や記述に，ユーザが日常生活で利用している状況を取り込む事で，ユーザは直感的な指定によりアクセス制御ができる．

また，情報家電環境は，パーソナルコンピュータやワークステーションなどの既存の計算機環境と比較し，異なるネットワークで接続された様々なハードウェア上で，多種多様なソフトウェアが動作するといった点において，ヘテロジニアスな環境であると言える．このヘテロジニア

スな環境に適応するため，本システムは，Java 言語を用いて記述されたミドルウェアとして動作し，分散システムに対するによるアクセス制御を提供する．従って，本システムは，オペレーティングシステムや単一の閉じたシステム内で提供されるアクセス制御に比べ，高い適応性を有する．

本システムでは，RUM サーバのアクセス制御機構を容易に利用できるライブラリを提供する．また，ポリシ設定ファイルにより，既存のポリシモジュールを組み合わせ，複雑なポリシの記述をプログラムを記述することなく行える．これらのライブラリの利用やポリシ設定ファイルの作成により，情報家電ソフトウェアの開発者は，少ない労力で環境情報適応型アクセス制御モデルを実現する情報家電アプリケーションを実装できる．

謝辞

本研究の機会を与えて下さり，絶えず丁寧なご指導を賜りました，慶應義塾大学環境情報学部教授徳田英幸博士に深く感謝致します．また，副査をしていただき貴重なコメントを下さった，慶應義塾大学環境情報学部教授清木康博士ならびに慶應義塾大学環境情報学部助教授楠本博之博士に深く感謝いたします．

修士論文の執筆にあたり貴重な助言を下さった，慶應義塾大学大学院政策・メディア研究科助教授戸辺義人博士，慶應義塾大学大学院政策・メディア研究科助教授西尾信彦博士，ならびに慶應義塾大学大学院政策・メディア研究科専任講師望月祐洋博士に深く感謝いたします．

慶応義塾大学徳田研究室修士課程 2 年中澤仁氏をはじめとする徳田研究室の諸先輩方に感謝します．

そして，研究室で共に執筆活動をした徳田研究室修士 2 年の皆さんに，この場を借りてお礼致します．

また，修士論文の執筆の後押しをしてくれた新田美智子氏や水野梨沙氏に大変感謝しております．そして，青木崇行氏や石井かおり氏，桐原幸彦氏，松宮健太氏をはじめとする徳田研究室の学生の皆さんに，この場を借りてお礼致します．

参考文献

[1] 楠本晶彦, 岩井将行, 中澤仁, 徳田英幸: 位置情報を用いたサービスの階層的検索と利用の実現, *DICOMO*, pp. 00–00 (2000).

[2] Kusumoto, A., Nakazawa, J., Tobe, Y. and Tokuda, H.: A Location-Adaptive Virtual Networked Appliance, *International Workshop on Smart Appliance and Wearable Computer*, pp. 214–219 (2001).

[3] 土田泰徳, 楠本晶彦, 大越匡 and 徳田英幸: 計算機の位置情報履歴を利用したサービス提供機構の構築, 第61回全国大会講演論文集 3-59 (2000).

[4] 桐原幸彦, 由良淳一, 楠本晶彦, 徳田英幸: 情報家電機器におけるプロアクティブな利用支援機構, 情報処理学会システムソフトウェアとオペレーティングシステム研究会 (OS研) 2001年6月 pp.105-112 (2001).

[5] IEEE: Standard for a High Performace Serial Bus (1999).

[6] Bluetooth SIG: The Official Bluetooth SIG Website.

[7] Sun Microsystems, Inc.: Jini Architecture Specification (1998). http://www.javasoft.com/products/jini/specs/jini-spec.pdf.

[8] Sony, Matsushita, Philips, Thomson, Hitachi, Toshiba, Sharp and Grundig: Specification of the Home Audio/Video Interoperability (HAVi) Architecture (1998). http://www.havi.org/home.html.

[9] Universal Plug and Play Forum: Universal Plug and Play (UPnP) (1999). http://www.upnp.org.

[10] Microsoft, Corp.: Universal Plug and Play Device Architecture Reference Specification (1999).

[11] Nakazawa, J., Okoshi, T., Mochizuki, M., Tobe, Y. and Tokuda, H.: VNA: An Object Model for Virtual Network Appliances, *International Conf. on Consumer Electronics*, IEEE (2000).

[12] 情報処理振興事業協会（ＩＰＡ）セキュリティセンター: 情報セキュリティの現状2000年版 (2001).

[13] Bellovin and Cheswick: Network Firewalls, *IEEE Communications*, pp. 50–57 (1994).

[14] Object Management Group: Common Object Request Broker Architecture and Specification (1996).

[15] Sun Microsystems, Inc.: Telecom Network Management With Enterprise (2001). http://java.sun.com/j2ee/white/nmEJBwp.pdf.

[16] Lai, C., Gong, L., Koved, L., Nadalin, A. and Schemers, R.: USER AUTHENTICATION AND AUTHORIZATION IN THE JAVA(TM) PLATFORM, *Published in the Proceedings of the 15th Annual Computer Security Applications Conference* (1999).

[17] Gong, L.: *Inside Java2 Platform Security*, Addison Wesley Longman, Inc. (2000).

[18] Samar, V. and Lai, C.: Making Login Services Independent From Authentication Techologies, *Proceedings of the SunSoft Developer's Conference* (1996).

[19] Miller, S. P., Neuman, C., Schiller, J. I. and Saltzer, J. H.: Kerberos Authentication and Authorization System (1988).

[20] Okoshi, T., Wakayama, S., Sugita, Y., Aoki, S., Iwamoto, T., Nakazawa, J., Nagata, T., Furusaka, D., Iwai, M., Kusumoto, A., Harashima, N., Yura, J., Nishio, N., Tobe, Y., Ikeda, Y. and Tokuda, H.: Smart Space Laboratory Project (2001).

[21] Sun Microsystems Inc.: Remote Method Invocation Specification (1996).

[22] World Wide Web Consortium: Extensible Markup Language(XML) 1.0 (1999).

[23] Wireless Mountain Laboratories, Inc.: The Spider - Tracking RF-identification technology into the Future.