

教育用 Windows PC を対象とした 教員が設定可能なアプリケーション実行制御システム

川上 崇^{†1} 河野 圭太^{†2} 山井 成良^{†2}

近年、大学等の教育機関で保有する教育用 Windows PC (教育用 PC) の数が増加しており、管理者負担軽減の目的から、雛形となる PC 環境を全ての教育用 PC に反映し、利用者に共通の環境として提供することが一般的になっている。その一方で、利用者の観点からは、各部局の都合や授業内容の都合に応じてアプリケーションの実行環境を変更できることが強く望まれており、管理者の要望とのトレードオフが大きな課題となっている。そこで本稿では、管理者による設定に加えて、教員によるアプリケーション実行制御の設定を実施可能とすることにより、管理者負担を増加させることなく、オンデマンドでアプリケーションの実行環境を変更できるシステムを提案する。

Teacher Configurable Execution Control System for Application Software on Educational Windows PCs

TAKASHI KAWAKAMI,^{†1} KEITA KAWANO^{†2} and NARIYOSHI YAMAI^{†2}

In recent years, the number of educational Windows PCs (educational PCs) in the organizations such as universities, has been increased. These educational PCs are generally managed all together for administrative reasons. One model image is distributed to all of the educational PCs, reducing the administrative tasks. On the other hand, from users' side, the right of executing certain application software should be granted according to the demand of divisions or classes. There is a trade-off between the request of administrators and users. This paper proposes a system to let teachers change the settings for execution control of application software. The proposed system changes the execution constraint of application software on demand, without increasing administrative tasks.

1. はじめに

近年、情報化の進展に伴い、大学等の学校教育において様々な教育分野の授業で PC を使う機会が増えている¹⁾。各教育分野に必要なアプリケーションは異なり、大学の都合、部局の都合、授業内容の都合に応じて個別のアプリケーション実行環境を提供する必要が出てきている。

例えば、全学的なセキュリティポリシーや部局独自で保有するライセンスの都合により、あるアプリケーションの利用を特定のアカウントや利用場所のみに制限したい場合がある。また、通常は利用が制限されているが、教員の指導の下である授業の間のみ利用を許可したいアプリケーションや、Web ブラウザなどの、授業中一時的に使用させたくないアプリケーションも

ある。そのため、任意のタイミングでアプリケーションの利用環境をアカウントごと、利用場所ごとに変更できることが求められている。しかし現実には、管理者負担の問題からこれらの要求に完全に対応することは難しい^{2),3)}。

この問題に対し、各教育用 PC 上でアプリケーションの起動制御を行うことで、アカウントごと、利用場所ごとのアプリケーション実行環境を提供するシステム (従来システム) が提案されている⁴⁾。従来システムでは、外部サーバにアプリケーションごとの起動制御条件を保持させ、教育用 PC に設置したアプリケーションの起動制御を実施するプログラムが、その外部サーバから能動的に取得した情報に基づいてアプリケーションの起動制御を行うことで、アカウントや利用場所などの状態に応じてアプリケーションの実行環境を変更できる。

しかし、従来システムでは、管理者があらかじめ設定しておいた起動制御条件に基づく実行環境の提供しかできなかったため、実際の利用状況にあわせて一時的に条件を変更したい場合などに対応することはできなかった。そのため、授業中に制御条件を変更し、それ

^{†1} 岡山大学 大学院自然科学研究科

Graduate School of Natural Science and Technology,
Okayama University

^{†2} 岡山大学 情報統括センター

Center for Information Technology and Management,
Okayama University

をリアルタイムに反映させるようなアプリケーション制御は実施できなかった。また、指定したアプリケーションを起動禁止にするブラックリスト方式での制御しか実施できなかったため、指定したアプリケーションのみを起動許可にするホワイトリスト方式での制御には対応できなかった。

そこで、本研究では、管理者の負担を増やさないという方針を保ちつつ、オンデマンドなアプリケーションの実行制御を行うために、教育用 Windows PC を対象とした教員が設定可能なアプリケーション実行制御システムを提案する。

提案方式では、大学などの管理運用体制の実状に合わせて、管理者による設定を教員が書き換えられる箇所と書き換えられない箇所を制御できるようにした。

提案システムは、制御情報を保持し、アプリケーションの利用可否を判断する利用判定サーバ、教育用 PC 上で実質的なアプリケーション実行制御を行うアプリケーション実行制御クライアント、利用判定サーバの制御情報を教員が授業中に変更するための制御情報変更クライアントの 3 つの要素から構成される。また、幅広いアプリケーション利用環境を提供するために、ブラックリスト方式でアプリケーションを制御する場合と、ホワイトリスト方式で制御する場合の 2 通りの制御を実現できるようにした。

2. 教育用 Windows PC の管理方法

一般的に大学などの組織で利用されている教育用 PC は管理者負担の問題から、イメージ配信方式やネットワークワークブート方式と呼ばれる方法で一括管理されている^{5),6)}。

イメージ配信方式では、あらかじめ用意しておいた雛形イメージを、PC を利用していない夜間などに各教育用 PC に反映させ、同一の利用環境として一括設定する。各教育用 PC に反映されたイメージは、次のイメージ配信時まで変わらないため、どのアカウントでログインしても利用できる PC 環境は固定される。そのためイメージ配信方式で、アカウントごとに異なる PC 環境を求めるといった要求に応えることは難しく、その代替案として各 PC ごとに異なるイメージを配信しておくという対応が考えられる。しかしこの場合用意するイメージの数だけセキュリティアップデートなどのメンテナンス作業を行わなくてはならなくなり、管理者の負担が増大してしまう。

一方でネットワークブート方式では、イメージ配信方式とは異なり、各教育用 PC の起動時に利用したいイメージを選択しダウンロードすることで、アカウントごとに異なる PC 環境を利用することができる。しかしこの方式でも、アカウントごとのイメージを作成しようとする、その数に応じてメンテナンスを行わなくてはならないイメージが増加するため、イメージ

配信方式と同様の管理者負担の問題が生じてしまう。

本稿で提案するシステムは、これら 2 つの PC 管理方法で一括管理されている PC 環境を前提としているが、岡山大学ではイメージ配信方式を採用していることもあり、以下では説明の都合上イメージ配信方式で管理されている教育用 PC を前提とする。

3. 従来のアプリケーション実行制御システム

3.1 概要

2 章で述べた問題を解決するため本研究グループでは、必要となるアプリケーションをすべて一つの雛形イメージにインストールしておくことで、イメージ管理に関する管理者の負担を軽減させつつ、各教育用 PC 上で個別の条件に応じてアプリケーションの起動制御を行うことで利用者に対する個別のアプリケーション利用環境を提供するシステム（従来システム）を提案している。

従来システムでは、教育用 PC 上でアプリケーションの起動を制御するアプリケーション制御プログラム（Application Control Program : ACP）を全ての教育用 PC に導入することによりアプリケーション利用の制御を行う。

ここで、Windows には、レジストリを統一管理することによって組織のルールに従った設定を組織内の全ての PC に適応させ、ユーザに PC をより安全にかつ便利に使用させるために利用するグループポリシーと呼ばれる機能がある⁷⁾。グループポリシーにはアプリケーションの起動を制御する機能があり、レジストリのキー値を設定することで管理者が指定したアプリケーションの起動を制限することができる。従来システムでは、この機能を用いて作り出したレジストリのデータファイル（.reg ファイル）をあらかじめ教育用 PC 内に用意しておき、必要ときにそのレジストリをインポートすることによってアプリケーションの起動制御を行う。

また、従来システムでは、アプリケーション起動可否の条件として、アカウント、IP アドレス、同時利用数を用いる。起動可否を判断するのは生徒側クライアントの ACP であり、判断材料となる起動制御の条件は外部サーバ（利用可否サーバ）が保持する。利用可否サーバは、ネットワークを通じて Windows にファイル共有やプリンタ共有などのサービスを提供する samba という機能を用いて制御情報を提供する⁸⁾。samba を用いた理由としては、複数の PC で情報を簡単に共有でき、また、最大同時接続数の制限機能を持っているため、同時起動数での制限も容易に実現できたからである。

具体的な動作としては、まずユーザが教育用 PC を起動すると ACP が samba で共有されているサーバ上のフォルダにアクセスする。アクセスするフォルダ

は管理されるアプリケーションごとに分けられており、アカウント名、IP アドレスでのみ制御されるものについては初期動作時にそれぞれのファイルを読み込む。ACP はそのファイルに自分のアカウント、IP アドレスがあれば許可、なければ禁止と自らが判断し、それぞれの状態になるように起動を制御する。同時接続数で制限されるものについては、初期動作後にユーザがボタンを押すというアクションを起こすことで、現在使用できる状態であるかどうかを確認する。

従来システムの構成と動作を図 1 に示す。

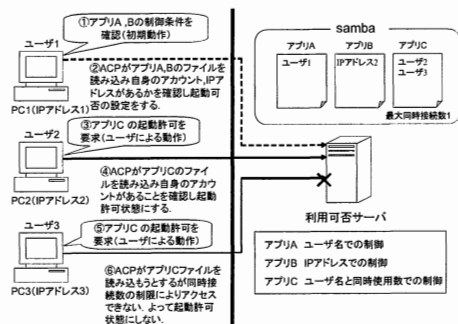


図 1 従来システムの構成と動作

Fig. 1 The structure and operation of the traditional system

図 1 ではアプリケーション A (アプリ A) はアカウントによる制御、アプリ B は IP アドレスによる制御、アプリ C はアカウントと同時利用数による制御を行っている。また、ユーザ 1 は教育用 PC 起動時の初期動作 (破線矢印) を、ユーザ 2, 3 は初期動作後のユーザによる起動許可要求の動作 (実線矢印) を行っている。まず、ユーザ 1 は初期動作としてアプリ A, B の起動可否を確認するために利用可否サーバの samba へアクセスする。アプリ A のファイルに書かれたアカウントと自身のアカウントが一致するのを確認して ACP はアプリ A を起動許可状態にし、アプリ B に書かれた IP アドレスと PC1 の IP アドレスは一致しないためアプリ B を起動禁止状態にする。ユーザ 2 は同時接続数で制御されるアプリ C の起動許可要求をユーザのアクションによって行い、自身のアカウントがファイルにあることを確認し、アプリ C を起動許可状態にする。その後、ユーザ 3 もアプリ C の起動許可要求をユーザのアクションによって行うが、samba の最大同時接続数の制限機能によりサーバへのアクセスができない。その結果 ACP は許可がおりなかったと判断し、アプリ C の起動許可を出さない。

3.2 問題点

従来システムの問題点としては、アプリケーション起動可否の条件をユーザ自身が問い合わせにくくプル方式を採用しているため、サーバ側からユーザに対し

て制御情報の変更を通知し、それに合わせて制御を行うような仕様にはなっていない。仮に制御情報の変更が通知されるようになっていたとしても、従来システムで制御するのは、アプリケーションの起動のみであるため、既に起動されているものに関しては禁止制御を実施しても、利用し続けることができる。

またアプリケーションの起動制御に必要なレジストリ情報は、グループポリシーエディタを通して得たものを、あらかじめ各教育用 PC に入れておかなければならないため、新たな制御対象アプリケーションの追加はイメージの配信を行う時にしか実施できない。

これらの問題から従来システムでは、オンデマンドでアプリケーションの実行環境を変更することはできず、授業内容に応じた制御は困難である。

加えて、すべてのアプリケーションが許可されている状態から、禁止するアプリケーションを追加していくことしかできないので、必要なアプリケーション以外はすべて起動禁止にするという要求に応えることはできない。アプリケーションは無数に存在することから、このシステムにおける制御条件の設定には限界がある。

4. 教員が設定可能なアプリケーション実行制御システム

4.1 実現方針

提案システムでは、オンデマンドでのアプリケーション実行環境の変更を可能とするために、まず新規の制御対象アプリケーションをシステム運用中に追加できるようにする。そして、制御対象となるアプリケーションの制御情報変更の権限を教員にも与え、その変更がリアルタイムで反映されるようにすることで、授業内容に応じた制御情報の設定を管理者負担を増やすことなく行えるようにする。ただし、部局の都合による制限がある場合など、教員に判断を委ねるべきではないアプリケーションについては、管理者の設定が最優先され、教員の設定変更は反映されないようにするなど、管理者と教員の役割を区別して設定できるシステムとして設計する。

上記のような教員が制御条件を設定可能なアプリケーション実行制御システムを実現するためには、従来システムの機能に加えて 3 つの機能が必要となる。まず、教員による利用可否サーバ内の制御情報の変更を実施する機能、次に、利用可否サーバ内の制御情報の変更に対し、制御条件を総合的に考慮して該当のクライアントへ通知する機能、最後に、制御クライアントが制御情報の変更を反映する機能である。なお、この機能追加により、利用可否サーバについては、利用可否の判定を実施することになるため、以後は利用判定サーバと呼ぶことにする。以下、3 つの機能の概要を述べる。

まず、教員が制御情報を変更するための機能は、変更した制御情報を利用判定サーバへ伝えるための、クライアントプログラムを作成することで実現する。そのプログラムでは教員のみが、変更する制御情報を利用判定サーバへ送信できるようにする必要がある。

次に、制御情報の変更に対し、制御条件を総合的に考慮してクライアントへ通知する機能は、利用判定サーバに、制御情報を格納したデータベースを保持させ、サーバ自身がその制御情報を考慮し、制御クライアントに制御情報変更の通知を行う機能を持たせることで実現する。提案方式における利用判定サーバでは、制御情報を従来のようにただ保持するのではなく、管理者と教員、双方の設定状況やアプリケーションの利用状況などを総合的に考慮して制御ユーザへ判定結果を通知できるようにする。

最後に、利用判定サーバから送られた制御情報を反映する機能は、アプリケーション制御プログラムに、サーバからの制御情報変更の通知を受信するための機能と、制御情報の変更により許可から禁止になったアプリケーションを強制終了する機能を持たせることで実現する。強制終了機能は3.2節で述べた起動中のアプリケーションに対する制御の問題に対応するために必要となる。

さらに幅広い制御条件の設定を実現するために従来のブラックリスト方式の制御だけでなく、指定したアプリケーション以外を実行禁止にするホワイトリスト方式の制御も行えるようにする。本稿ではブラックリスト方式で制御を行うときの状態をデフォルト許可状態、ホワイトリスト方式で制御を行うときの制御状態をデフォルト禁止状態とする。以下これら2つの状態をまとめてデフォルト状態と表現する。

これらの方針に従い、教員が制御条件を設定可能なアプリケーション実行制御システムを実現する。

4.2 システム構成

提案システムも、外部サーバの情報に基づき、各教育用PC上でアプリケーションの制御を行うことで個別のアプリケーション利用環境を提供するという基本的な構成は、従来システムと同様である。しかし、提案システムでは、従来システムのプル方式によるアプリケーション制御だけでなく、教員による設定変更をリアルタイムで反映させるプッシュ方式によるアプリケーション制御を可能とする。

提案システムの構成と動作を図2に示す。

図2に示すように、組織内の教育用PCに実行制御プログラムを導入し、教員が利用するPCには制御情報変更プログラムを導入しておく。利用判定サーバは制御情報を格納するデータベースを持ち、クライアントからの要求に対し、制御情報を総合的に考慮して応答する。教員による制御情報の変更は、制御情報変更プログラムから利用判定サーバへ変更の依頼をし、利用判定サーバが依頼内容と、制御情報を考慮して該当

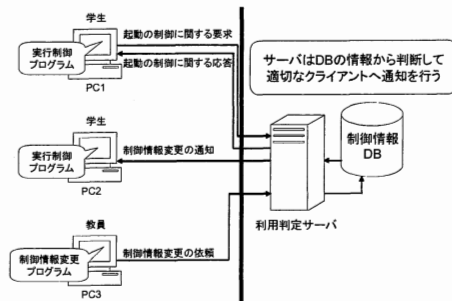


図2 提案システムの構成と動作

Fig. 2 The structure and operation of the proposed system

クライアントへ、制御情報の変更を通知する。

実行制御プログラム、制御情報変更プログラム、利用判定サーバの3つの要素の詳細を以下に示す。

アプリケーション実行制御プログラム（実行制御プログラム）

このプログラムは、従来システムと同様に各教育用PC上で実際にアプリケーションの制御を行うプログラムである。3.2節で述べた新規制御対象アプリケーション追加の問題に対応するために、アプリケーションの起動制御の方法は、従来の教育用PC上に用意されたレジストリファイルをインポートする方式から、サーバから制御に必要なデータを受信し、それを直接プログラムからレジストリに書き込む方式に変更する。ログイン時には、アカウント、ログインした教育用PCのIPアドレスを利用判定サーバに提出し、制御対象アプリケーションの情報（レジストリ情報、プロセス名）、各アプリケーションの利用可否、デフォルト状態を受け取り、これらの情報に従ってアプリケーションの起動を制御する（初期状態）。また、同時起動数による制御が必要になるアプリケーションについては、実際の利用時に起動可否を問い合わせ、その指示に従った制御をする機能を持つ。さらに、教育用PC利用中に、教員による制御情報の変更があれば、利用判定サーバから制御情報変更の通知を受け取って、その通知内容に従ってアプリケーションを制御する設定を追加する。その際、既に起動しているアプリケーションが禁止設定に変更された場合には、そのアプリケーションを終了させなければならないので、終了を促すウィンドウを表示させた後、それでも終了されない場合には強制終了する機能も追加する。このプログラムは、アプリケーションの起動を直接制御するため、全ての教育用PCで常駐させる。

教員用制御情報変更プログラム（制御情報変更プログラム）

このプログラムは教員が利用判定サーバの制御情報の

操作を行うためのプログラムである。教員にのみ使用させるためアカウントとパスワードを利用判定サーバへ送信し認証させる。アプリケーション個別の制御情報の変更とデフォルト状態の変更を行うことができ、アプリケーションごとに制御対象アカウント、IP アドレス、最大同時利用数などを設定することで、幅広い制御条件を指定することができる。ログアウト時には、自動で利用判定サーバへ教員がログアウトした旨を通知し、教員により変更された制御情報が残らないようにする。後述するように、教員が変更できる制御情報は管理者が教員による設定変更を許可した範囲に限られる。

アプリケーション利用判定サーバ（利用判定サーバ）
このサーバは従来の制御情報だけを提供し、アプリケーション利用可否の判断をクライアントに委ねるといった仕様から、制御情報から、総合的に考慮し応答、通知を実施する仕様に変更する。具体的には各教育用 PC のアプリケーション制御情報を格納するデータベースを持たせ、データベース内には管理者が設定するテーブル、教員が設定するテーブル、クライアントのログイン状態を記録するテーブルを用意し、通知先や通知する内容はそれらのテーブルを総合的に考慮し送信する。

前述のように大学などの管理・運用体制の実状により、アプリケーションによっては管理者による制御条件の設定のみが反映されるべきものもあるため、制御条件の初期設定は、管理者が行う。教員による制御情報の変更を許可するアプリケーションについては、管理者が設定するテーブル内に変更を許可する範囲（ユーザもしくは IP アドレス）と共に設定する。ここでは、教員が何も設定していないときの制御条件（基本許可、禁止）を管理者が設定できるようにする。

4.3 動作手順

システムの動作手順を図 3 に示す。図 3 では、まず学生が教育用 PC にログインし、制御対象アプリケーションの情報（レジストリ情報、プロセス名）、各アプリケーションの利用可否、デフォルト状態などのログイン時の初期状態を要求し、利用判定サーバからそれらの情報を受け取り、反映する初期動作を示す。そして教員が制御情報変更プログラムを用いて制御条件を変更した場合に、それが学生の教育用 PC に反映されるまでの動作を示す。

図 3 における動作手順を以下にまとめる。
（学生が教育用 PC にログインする）

- (1) 実行制御プログラムは初期状態の情報を得るために利用判定サーバへアカウントと IP アドレスを送信する。
- (2) 利用判定サーバは制御情報データベースにクライアントのログイン情報（アカウント、IP アド

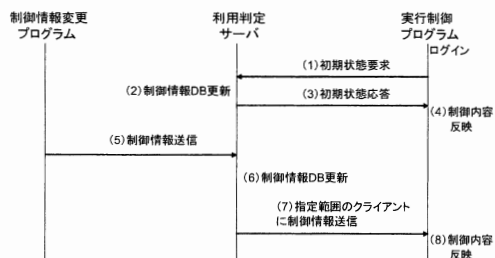


図 3 動作手順

Fig. 3 The operation process

レス)を追加する。

- (3) 利用判定サーバは制御情報 DB の内容に基づき、そのアカウント、IP アドレス、教員制御などの情報を考慮した初期状態を対象クライアントへ返す。
- (4) 実行制御プログラムは受信した初期状態の情報に基づき制御を実施する。
（教員が制御情報変更プログラムにログインし、情報を変更する）
- (5) 制御情報変更プログラムは教員が変更した情報を利用判定サーバへ送信する。
- (6) 利用判定サーバは制御情報データベースに教員による制御情報を追加する。
- (7) 利用判定サーバは変更に該当するクライアントへ制御情報変更を通知する。
- (8) 実行制御プログラムは受信した制御情報の変更内容に基づき制御を実施する。

5. 提案システムの実装

5.1 アプリケーション実行制御プログラム

C#を用いて制御条件変更に対応したアプリケーション実行制御プログラムを作成した。

まず、アプリケーションの起動制御は、従来と同様にグループポリシーの機能を応用して実現した。グループポリシーでは、実行ファイルのハッシュ値による起動制御と、パス指定による起動制御を行うことができる。個別のアプリケーションの起動制御は、対象となるアプリケーションの実行ファイルのハッシュ値とファイルサイズを特定のレジストリファイルに書き込むことで行われる。提案システムでは、3.2 節で述べた新規制御対象アプリケーション追加の問題に対応するために、ハッシュ値、ファイルサイズ、キー値といったレジストリ情報を自作したプログラムから計算できるようにした。そしてその計算結果を管理者が利用判定サーバに登録できるようにし、その登録したレジストリ情報をサーバから受け取り、クライアントプログラムから直接書き込むことにより制御を行うようにした。図 4 にアプリケーション個別の制御を行う際

のレジストリファイルの設定例を示す。

名前	種類	データ
(既定)	REG_SZ	(値の設定なし)
HashAlg	REG_DWORD	0x00008003 (32771)
ItemData	REG_BINARY	b6 0d d4 d2 06 3e e4 1c b8 e4 87 fc fb b6 41 9e
ItemSize	REG_DWORD	00 bf 09 00 00 00 00 00
ProcessName	REG_SZ	explore
SoftwareName	REG_SZ	Internet_Explore

図 4 レジストリファイルの設定例

Fig. 4 A setup example of registry file

制御に必須のデータは HashAlg, ItemData, ItemSize の3つである。HashAlg はハッシュアルゴリズムのことで実行ファイルのハッシュの取得に MD5 を用いるということは既定なのでこの値は 32771 (10 進) で固定である。ItemData には実行ファイルのハッシュ値, ItemSize にはファイルサイズをそれぞれ入れる。これらの情報を設定例のようにレジストリに書き込むことで、アプリケーションの起動制御を行うことができる。

デフォルト状態はグループポリシーの、パスを指定することにより起動を制御できる機能とファイルのハッシュ値を指定することにより起動を制御できる機能を組み合わせて実装した。デフォルト禁止状態は、起動を禁止するパスの指定範囲をワイルドカード文字のアスタリスクに設定することですべてのアプリケーションの起動を禁止し、例外的に許可するものに関しては、起動を許可するファイルのハッシュ値を指定することにより起動制御できるようにした。設定した規則の優先度はより具体的に指定されたものが優先されるので、ハッシュ値で許可と指定したアプリケーションはパスで指定した禁止規則よりも優先されて起動を許可される⁹⁾。

また、サーバとのやりとりは TCP ソケット通信を用いて行う。ログイン時には、4.3 節で述べた初期状態に必要な情報を受け取り、教員による制御情報変更時には、対象となるアプリケーションのプロセス名を受け取る。制御情報の変更に対応するため、実行制御プログラムは PC ログイン時から、常に listen 状態で待機するようにした。同時利用者数によって制御されるアプリケーションについては、ユーザが自ら起動可否の許可を問い合わせることで利用できるかどうかを確認することができ、その際には判定結果を受け取り、その結果に基づき制御を行うようにした。

さらに、起動中のアプリケーションが制御情報の変更により、使用が禁止された場合に必要となる、アプリケーションの強制終了機能は、まず、C#のクラスライブラリに用意されている、実行されている全てのプロセスを取得するメソッドを用いて起動プロセスを読み取り、プロセス名が一致するものを見つけ出す。そしてその一致したプロセスに対し、まず終了を促す

メッセージを表示させ、一定時間経過後終了されていなければ close 命令、さらに一定時間経過後終了されていなければ kill 命令を実行する。close 命令だけであると、例えばテキストエディタで内容の変更を保存していなかった場合などに、保存を促すダイアログが表示され、アプリケーションが確実に終了されない場合がある。そのような場面を想定し、close 命令実行後、まだプロセスが残っている場合には、kill 命令という確実にアプリケーションを終了させる命令を実行する。

図 5 にアプリケーション起動制御プログラムの実行画面を示す。

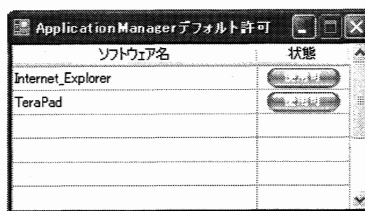


図 5 アプリケーション実行制御プログラム

Fig. 5 The application control program

図 5 ではアプリケーション制御プログラムの基本状態の画面が示されている。同時接続数で制限されているアプリケーションを利用する際に行う起動可否の問い合わせは、状態と書かれた列にあるボタンを押すことで実施でき、従来のプル方式による起動許可要求の機能も備えている。タイトルバーには現在のデフォルト状態が表示されるようにした。

5.2 教員用制御情報変更プログラム

C#を用いて教員による制御情報の変更を行うことができるクライアントプログラムを作成した。

制御情報変更プログラムからは変更の内容を利用判定サーバへ送信するのみで、対象となる教育用 PC への通知は利用判定サーバによって行われる。利用判定サーバとのやりとりは TCP ソケット通信を用いて行い、教員以外のユーザに操作させないために最初の認証でユーザ名とパスワードを送信し、このとき確立したコネクションを用いて変更する制御情報を送信するようにした。

図 6 に制御情報変更プログラムの画面を示す。

制御情報変更プログラムの各機能はタブで分割し、最初にログイン画面でユーザ認証を行わないと制御情報の変更はできないようにした。図 6 ではアプリケーション制御タブが選択されており、ここでは各コンボボックス、テキストボックスに設定するパラメータを入力し、制御情報追加ボタンを押して利用判定サーバへ制御情報追加の要求を送信する。制御情報の追加が成功すると追加ボタンの下のリストボックスに教員ユーザが追加した制御情報が書き込まれ、その書き込まれ

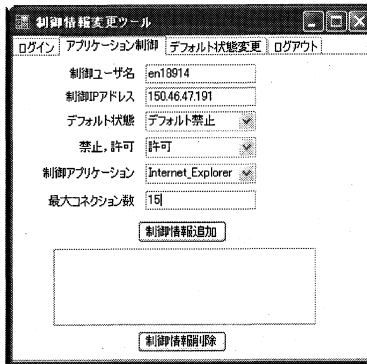


図 6 制御情報変更プログラム

Fig.6 The program to change control information

た情報を選択し、削除ボタンを押すと利用判定サーバへ制御情報の削除の要求を送信する。

5.3 利用判定サーバ

利用判定サーバは perl で、アプリケーションの制御情報とクライアントと教員のログイン状況を格納するデータベース（制御情報データベース）は MySQL を用いて作成した。制御情報データベースには、管理者の制御情報を書き込む管理者用テーブル、教員の制御情報を書き込む教員用テーブル、個別のアプリケーションに関するテーブル、クライアントのログイン状況を書き込む利用者テーブルを用意した。

まず、起動制御プログラムからの問合せに対する起動許可判定は、起動制御プログラムから受信したアプリケーション名を元に制御情報データベース内を探索し、クライアントのログイン情報と制御内容を照合して、判定した結果を応答するように実装した。

次に、教員による制御情報の変更は、制御情報変更プログラムによる認証完了後、教員用テーブルの書き換えを実施することで可能とした。教員の設定変更を反映してもよいかどうかを判断するために、教員からの制御情報変更の要求が届くとまず、管理者用の対象アプリケーションに対する設定テーブルを参照する。管理者用の設定テーブルでは、管理者による起動許可、禁止（教員による変更は不可）に加え、教員の設定を反映させるが、教員の指示がない場合基本許可、禁止の4状態を設定可能とした。管理者用のテーブルの設定が、基本許可、禁止のいずれかだった場合、教員用テーブルに制御情報変更の内容を書き込むようにし、利用可否の判定時に参照されるようにした。なお、今回は、教員が自身の講義とは無関係な設定をすることや、複数の教員間で相反する設定をすることは想定しておらず、仮にそのような状況が生じた際には、先に設定した内容が反映される実装となっている。

制御情報が変更になった場合、クライアントのログイン情報を参照して制御対象となる利用者に制御内容を変更したことを伝えるようにした。また、利用判定

サーバは各アプリケーションの現在の利用数を把握するために、アプリケーションの許可判定時に許可が出たときはその利用者の情報を対象アプリケーションのテーブルに記録するようにし、このテーブルに記録されている利用者の数を用いることにより、アプリケーションの同時起動数による制御を実現した。

6. 動作確認実験

今回実装した試作システムを用いて動作確認実験を行なった。まず、PC1, PC2, PC3, PC4の4つのPCを用意し、PC1, PC2に実行制御プログラム、PC3に制御情報変更プログラムを導入し、PC4を利用判定サーバとした。PC1のIPアドレスを192.168.1.11、PC2のIPアドレスを192.168.1.12とし、それぞれのPCを利用するユーザアカウントをuser1, user2とした。また、実行制御プログラム、利用判定サーバ、制御情報変更プログラムが通信可能なネットワーク環境を構築し、制御の対象となるアプリケーションとしてInternet Explorer, TeraPad, Excelを使用し、以下に述べる3つの実験を行なった。

- **実行制御プログラムからの問い合わせによる制御**
この実験では、従来から実装されているユーザからの問い合わせに対する利用可否の返答を正しく行えるか、また同時利用数による制御が正しく行われるかを確かめた。また、今回新たに実装したデフォルト状態が各教育用PC上で機能しているかを確かめた。

[制御情報の設定]

管理者による設定で user1 は Internet Explorer の使用を禁止、192.168.1.12 は TeraPad の使用を禁止、Excel は誰が使用してもよいが同時利用数は1とした。また user1 はデフォルト許可状態、user2 はデフォルト禁止状態に設定した。

[実験手順]

- (1) PC1, PC2 でアプリケーション実行制御プログラムを起動した。
- (2) PC1 から Internet Explorer と Excel の使用許可を求める問い合わせをした。
- (3) PC2 から TeraPad と Excel の使用許可を求める問い合わせをした。

[実験結果]

PC1 (user1) はアカウントによる制御により Internet Explorer の使用許可はおりず、Excel の使用許可のみおりることを確認した。また、その他のアプリケーションは全て使えるデフォルト許可状態になっていることを確認した。

PC2 (192.168.1.12) は IP アドレスによる制御により TeraPad の使用許可はおりず、同時利用数の制御により、Excel の許可もおりなかった。また、その他のアプリケーションも全て使えないデ

フォルト禁止状態になっていることを確認した。

● 制御情報変更に対する制御情報の反映

今回授業中に教員が特定のアプリケーションの使用を禁止に設定変更することを想定し、実装した機能が正しく動作するかを確かめた。また既に起動しているアプリケーションが教員により、使用禁止に設定変更された場合に、終了処理が正しく行われるかを確かめた。さらに今回新たに実装した、デフォルト状態の機能が、教員の設定変更によって正しく切り替えられるかを確認した。

[制御情報の設定]

管理者による設定で user1 は Internet Explorer の使用を基本許可とした。またデフォルト状態は許可状態とした。

[実験手順]

- (1) PC1 でアプリケーション実行制御プログラムを起動した。
- (2) PC1 で Internet Explorer を起動した。
- (3) PC3 で教員用制御情報変更プログラムを起動した。
- (4) PC3 からユーザ1 が Internet Explorer を使用禁止となるように設定した。
- (5) PC3user1 のデフォルト状態をデフォルト禁止状態に設定した。

[実験結果]

教員用制御情報変更プログラムで、アプリケーションに関する制御情報変更の設定が行うと、PC1 に通知が届き、起動していた Internet Explorer が終了され、起動禁止状態になることを確認した。さらにデフォルト状態に関する制御情報変更の設定を行うと、PC1 に通知が届き、デフォルト禁止状態となることを確認した。

以上の結果より、アプリケーション起動制御プログラムが利用判定サーバの設定どおりに制御を行うこと、制御情報変更プログラムからの個別のアプリケーションに関する制御情報の変更と、デフォルト状態に関する制御情報の変更が、制御対象クライアントにリアルタイムで反映されることを確認した。

7. む す び

本稿では、教員が設定を変更できるアプリケーション実行制御システムを提案し、設計、実装と動作確認実験を行った。

アプリケーションの起動を制御する従来のシステムでは、利用判定サーバからアプリケーション制御クライアントに制御情報の変更を伝える仕組みが無かったため、制御情報を運用途中で変更し、反映させることは不可能であった。そこで提案システムでは、利用判定サーバにログインユーザの情報と、制御アプリケーションの情報を把握し、制御情報の変更を通知する機

能を持たせ、さらに制御クライアント側にも、アプリケーションの強制終了機能、制御情報変更通知の待ち受け機能を持たせることで、教員による制御情報の変更をリアルタイムに反映させることを可能とした。このことにより管理者の負担を増やすことなく、従来システムのような静的な制御だけでなく、授業に応じたアプリケーション制御のような動的な制御を行えるようになった。

今後の課題として、実環境での動作確認を行なうための提案システムの負荷テスト、管理者による制御情報の登録を簡素化できるツールの開発、またシラバスシステムなどとの連携により、管理者の負担を増やすことなく教員が制御できる範囲を適切に限定できる機能の追加などが挙げられる。

参 考 文 献

- 1) 文部科学省, “学校における教育の情報化の実態等に関する調査結果,” <http://www.mext.go.jp/b.menu/houdou/21/08/1283617.htm>.
- 2) 今井晴基, 村瀬正名, 杉山潤, 根岸康, “オーブソンス・ソフトウェア spcman を用いたクラスルーム PC 管理,” 情報処理学会論文誌, Vol.2005, No.104, pp.23-28, Oct. 2005.
- 3) 庄司文由, “教育用 PC 端末郡の PC クラスタの利用とその運用について,” 情報処理学会論文誌, Vol.2004, No.77, pp.65-69, July 2004.
- 4) 関谷章仁, 川上崇, 河野圭太, 山井成良, “教育用 WindowsPC における同時起動数を考慮したアプリケーション制御システム,” 情報処理学会研究報告, Vol.2010, No.1, pp1-6, Mar. 2010.
- 5) 堀亜砂美, 横山節雄, 宮寺庸造, “私立大学における情報環境の考察,” 情報処理学会論文誌, Vol.2007, No.12, pp.61-68, Feb. 2007.
- 6) 奥村勝, 藤村丞, “1000 台規模のディスクレス PC システムの構築と運用,” 情報処理学会研究報告, Vol.2008, No.23, pp.61-66, Mar. 2008.
- 7) Microsoft Windows Server, “グループポリシー管理コンソールによる企業システムの管理,” <http://www.microsoft.com/japan/windowsserver2003/gpmc/default.mspx>.
- 8) 日本 Samba ユーザ会, “Samba max connections,” <http://www.samba.gr.jp/project/translation/2.2.5/manpages/smb.conf.5.html#MAXCONNECTIONS>.
- 9) Microsoft TechNet, “Windows XP クライアントのソフトウェア制限ポリシー,” <http://technet.microsoft.com/ja-jp/library/dd347746.aspx>.