アプリケーション実行制御システムにおける教員による講義中での制御対象変更機能

藤原正憲[†1] 川上崇[†1]
河野圭太[†2] 山井成良[†2]

本研究グループでは，教員が講義中に学生のアプリケーション利用環境を制御することを目的として教育用 Windows PC を対象としたアプリケーション実行制御システム（従来システム）を開発してきた．従来システムでは，制御対象とすることのできるアプリケーションは事前に管理者が登録する必要があった．そのため，教員が望んだアプリケーションが制御対象として登録されていない場合，教員はそれを制御対象として管理者に登録してもらい，教員・学生のプログラムを再起動する必要があった．また，全てのアプリケーションの起動を禁止とするデフォルト禁止状態に設定を変更しても既に起動中のアプリケーションは強制終了できなかったため，学生はそのままアプリケーションを使用可能であった．これらの問題を解決するため本稿では，教員が制御対象アプリケーションをシステム起動中に登録する機能および設定をデフォルト禁止状態に変更した際に起動中のアプリケーションを強制終了する機能を提案する．

On-demand Configuration Feature of Target Programs Manageable for Teachers on Application Execution Control System

Masanori Fujiwara,[†1] Takashi Kawakami,[†1] Keita Kawano[†2] and Nariyoshi Yamai[†2]

Our research group has developed an on-demand application execution control system on the educational Windows PC (traditional system). In the traditional system, control target application should be registered by the administrator beforehand. When the application which the teacher wanted to manage was not listed in the control target, the application had to be registered and the programs had to be restarted.Moreover, the traditional system cannot kill the whole running application software when the default rule is changed into prohibition. Therefore, students could continue to use the application. In order to solve these problems, this paper proposes two features. One is the feature that lets teachers add application software to the control list while our system is in service. The other is the feature that kills all application software the student has used when a teacher changes the default rule into prohibition.

1. はじめに

近年，情報化が急速に進展したことを背景とし，情報教育の必要性が高まっている[1)]．そのため，学校教育において PC を活用する機会が増え，大学などの組織が保有する教育用 PC の数が増大している．このような教育用 PC の運用にあたり，複数の施設に教育用 PC を分散的に配置する一方，利用者がどこの施設で PC を利用しても同一の環境を利用できるように一括管理することが一般的である[2),3)]．

このような環境を効率よく実現するために，大学などではあらかじめ作成した雛形イメージをコピーし，全ての教育用 PC に反映させるイメージ配信方式や，教育用 PC の起動時に利用するイメージをダウンロードするネットワークブート方式などの方式が用いられている． しかしながら，これらの方式を用いて，授業の内容やライセンス数の都合などにより，アカウントや利用場所に応じて学生に個別のアプリケーション利用環境を提供しようとすると，多数のイメージが必要となり，管理コストの増加が問題となる．そのため，管理コストを大きく増やすことなくこのようなアプリ

†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

ケーション利用環境を提供する方法を検討する必要がある．

このような要求に応えるため，我々の研究グループでは，管理負担を必要以上に増やさないという方針の下，利用者のアカウントや利用する場所などに応じて個別のアプリケーション利用環境を提供することを目的として，アプリケーション実行制御システム（従来システム）を開発してきた[4),5)]．このシステムでは教員が制御対象とする学生または PC とアプリケーションを指定して個別アプリケーションの制御情報をリアルタイムに変更することや，原則として全てのアプリケーションを禁止，または許可とするデフォルト禁止・許可の切り替えを行うことができる．

しかし，従来システムでは制御対象アプリケーションとして指定できるアプリケーションはシステムの管理者のみしか登録することができなかった．また，管理者が新しいアプリケーションを制御対象として登録しても，既に起動中の学生と教員それぞれのプログラムにその情報が反映されなかった．そのため，もし教員が授業中に制御対象としたいと考えたアプリケーションが制御対象アプリケーションとして事前に登録されていなかった場合，管理者に問い合わせて制御対象アプリケーションとして登録してもらい，教員用 PC，学生用 PC を共に再起動する必要があり，現実的ではなかった．

さらに，従来システムではデフォルト状態を禁止に切り替えたときにアプリケーションを強制終了する機能が実装されていなかった．そのため，デフォルト禁止に指定されたユーザでもすでに起動中のアプリケーションは強制終了されず，そのまま使用可能であった．

そこで本研究では，これらの問題を解決するため，教員自身が講義中でも自身のプログラムから容易に制御対象アプリケーションを追加登録できる機能およびデフォルト禁止への切り替え時にユーザが起動しているアプリケーションをすべて強制終了させる機能を提案する．

2. 従来のアプリケーション実行制御システム

2.1 教育用 PC の一括管理とその問題

大学などで用いられる教育用 PC は様々な場所に置かれている教育用 PC の中から学生がどの PC を用いても同じような環境を利用できるようにイメージ配信方式やネットワークブート方式などの方式を用いて管理することが一般的である．

イメージ配信方式では，あらかじめ雛形となるイメージを作成しておき，そのイメージを教育用 PC が利用されない夜間などに各教育用 PC へ反映させることで同一の PC 利用環境となるように設定する．この方式でイメージを反映した PC ではどのアカウントを用いてログインをしても，利用することのできる PC のアプリケーション利用環境は固定のものとなるため，アカウントごとや，授業中に授業の状況に応じてといったアプリケーション利用環境の変更は困難である．また，教室ごとなどの利用場所ごとに異なるイメージを反映することで異なったアプリケーション利用環境を構築することは可能であるが，必要となるイメージ数が増加してしまい，それぞれのイメージに対するメンテナンスのための管理負担の増加という問題が生じる．

一方で，ネットワークブート方式では教育用 PC の起動時に利用したいイメージを選択してダウンロードするという方法をとる．この方式は，イメージ配信方式とは異なり，起動時に利用するアプリケーション利用環境が反映される．そのため，起動時にダウンロードするイメージを選択することでアカウントごとに異なるアプリケーション利用環境を構築することができる．しかしながら，この方法でもイメージ配信方式と同様に授業ごとやアカウントごとに専用のイメージを作成することは管理負担の面から難しく，授業中にアプリケーション利用環境を変更することも困難である．

2.2 従来システムの概要

2.1 節で述べた問題を解決するために本研究グループでは，管理負担を必要以上に増やすことなく授業状況や利用場所に応じて学生のアプリケーション利用環境の変更を実現することを目的としてアプリケーション実行制御システムを開発してきた．

まず，従来システムの構成および動作を図 1 に示す．

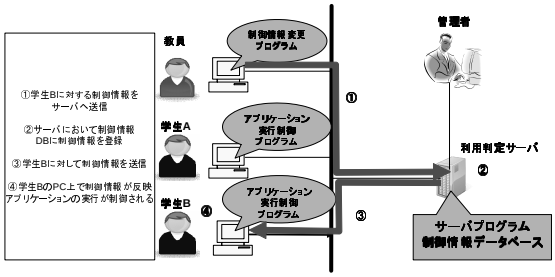

図 1 従来システムの構成と動作
Fig. 1 The structure and operation of the traditional system.

従来システムは教員が用いて学生 PC 上のアプリケーションに対する制御情報を設定する教員用制御情報変更プログラム，実際に学生 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実行制御プログラム，学生 PC 上のアプリケーションの利用を判定する利用判定サーバプログラムから構成される．また，サーバプログラムを実行する PC にはアプリケーションの制御情報を保持するデータベース（以後，制御情報 DB）を置く．

動作としては，図 1 に示したように教員が制御情報

変更プログラムを用いて設定した制御情報を一旦サーバプログラムへ送り制御情報 DB へと登録する．その後，サーバプログラムからアプリケーション実行制御プログラムへ制御情報を送信し，アプリケーション実行制御プログラムが制御情報を受け取り，実際に学生 PC 上のアプリケーションを制御する．

以後，それぞれのプログラムおよび制御情報 DB の詳細について述べる．

- **教員用制御情報変更プログラム**
  教員用制御情報変更プログラムでは，学生または学生が使用している PC のどちらかを指定してデフォルト状態または個別のアプリケーションに対する制御情報の変更を行うことができる．ここでいうデフォルト状態とは学生 PC 上全てのアプリケーションに対する制御であり，デフォルト禁止状態に指定された学生は全てのアプリケーションの実行ができなくなる．制御情報を変更する対象となる学生の指定には対象学生のユーザ名，もしくはそのユーザが使用している PC の IP アドレスを用いて個別にユーザを指定する他，サブネット単位で指定することで，演習室単位で制御することもできるようにしている．
  個別アプリケーションの制御では，ユーザの指定後，制御対象アプリケーションリストの中から制御対象とするアプリケーションを選択して，そのアプリケーションを禁止とするか許可とするかを選択して制御情報をサーバへ送信する．これに加えて，このときライセンスの都合などで利用許可を出すユーザ数を制限するために，最大同時接続数を設定することもできる．
  一方，デフォルト状態の変更においては個別アプリケーションの制御と同様に制御情報を変更するユーザを指定した後，デフォルト状態を許可とするか禁止とするかを選択して情報をサーバへ送信する．
  また，教員が PC からログアウトした際には，授業が終了したとみなしてそれまでログインしていた教員が登録していた制御情報を全て削除する．
  このプログラムの実装には C#を用いている．
- **アプリケーション実行制御プログラム**
  アプリケーション実行制御プログラムはユーザの PC 上で起動し，ユーザ PC 上のアプリケーションの実行を実際に制御するプログラムである．このプログラムは起動時に，このプログラムを起動している PC の IP アドレスとユーザ名を取得して，それをサーバへ送信することで，これらの情報を後述する制御情報 DB 中のクライアントリストに登録する．その後，サーバから制御対象アプリケーションとその時点でアプリケーションが使用許可か不許可かを取得し，このプログラム中の制御対象アプリケーションリストに反映する．そして，このプログラムの起動中に制御情報変更プログラムからサーバを経由して制御情報の変更を受信すると制御情報が反映され，アプリケーションの制御を行う．
  アプリケーションの制御は，Windows のグループポリシーの機能を用いて実施している．[6),7)] 個別のアプリケーションに対する制御は，対応するレジストリに制御したいアプリケーションのハッシュ値，ファイルサイズを書き込むことで制御を行う．
  一方，デフォルト状態に対する制御はハッシュ値のかわりにアプリケーションのパスをワイルドカード文字である*としてレジストリに書き込むことで実現している．
  また，このプログラムはもし個別に学生 PC 上で起動中のアプリケーションが禁止に指定された場合，このアプリケーションを強制終了させる．この強制終了は，ユーザが実行しているアプリケーションのプロセスを検出し，禁止に指定されたアプリケーション名とプロセス名が一致した場合に指定時間強制終了まで作業の保存のための猶予を持たせた後，そのプロセスに対して kill 命令を出すことで実現している．
  このプログラムの実装には C#を用いている
- **サーバプログラム**
  サーバプログラムは，制御情報 DB を持つマシン上で起動し，教員用制御情報変更プログラムから受け取った制御情報を DB に登録し，その情報を対応するユーザの制御情報変更プログラムへと送信する．また，ユーザのアプリケーション実行制御プログラムの起動時などにアプリケーション利用可否の判定の問い合わせがあった場合，制御情報 DB の制御情報の登録状況に応じた判定結果をアプリケーション実行制御プログラムへと返す．
  このプログラムの実装には Perl を用いている．
- **制御情報 DB**
  制御情報 DB は，アプリケーションの制御情報や，ユーザの情報を格納し，この制御情報 DB の内容を元にアプリケーションの利用可否を判定する．
  制御情報 DB には，図 2 に示すような 7 種類のテーブルが定義されている．以後それぞれのテーブルの役割について詳細に述べる．
  制御対象アプリケーションテーブルには制御対象とするアプリケーションのアプリケーション名，前述したハッシュ値，ファイルサイズ，書き込んだレジストリ情報の識別子となる key をレコードとして登録するようにしている．このテーブルの中にあるアプリケーションが教員が制御対象とできるアプリケーションとなる．
  クライアントリストテーブルは，アプリケーション実行制御プログラムを起動中の PC の IP アド

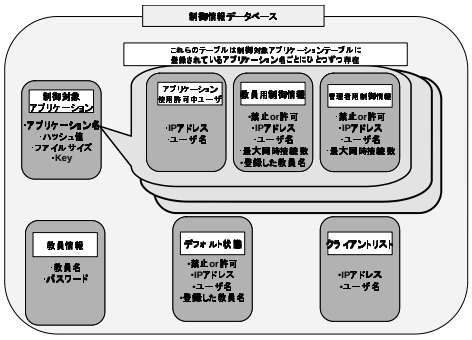

図 2 制御情報データベース
Fig. 2 Control information database.

レスとその PC にログインしているユーザ名をレコードとして持つ．このテーブルは，教員が制御情報を変更する際に指定した制御対象ユーザがログイン中かどうかの識別に用い，ログイン中と判定されれば，制御情報の送信を行う．

デフォルト状態テーブルは，デフォルト禁止か許可かという制御情報，制御情報を適用するユーザの IP アドレスやユーザ名を入れるレコードとデフォルト状態を指定した教員名をレコードとして持つようにしている．教員名は教員がログアウトした際にこの制御情報を削除してよいかどうかを識別するために用いる．このテーブルを基に学生のデフォルト状態が許可か禁止かを判定する．

教員情報テーブルは教員名とその教員のログインに使用するパスワードをレコードとして持つ．そして，教員のログイン時にログイン可否の判断に用いる．

制御情報テーブルは管理者用のものについては禁止か許可の制御情報，制御対象ユーザの IP アドレスおよびユーザ名，最大同時接続許可数をレコードとして持たせている．また，教員用制御情報テーブルは管理者用のテーブルと同様のレコードに加えて教員のログアウト時にどの情報を削除するかの識別をするために制御情報を登録した教員名を入れるレコードを持たせている．これらのテーブルを基にアプリケーションの利用可否判定を行う．利用可否判定は，管理・運用の都合上，管理者のみが設定を行うべきアプリケーションもあるため，管理者用制御情報テーブルの内容を教員用制御情報テーブルの内容より優先する様にしている．また，教員用制御情報テーブル内の制御情報については，後から登録されたものを最新の制御情報とみなして優先するようにしている．

アプリケーション使用許可中ユーザテーブルには IP アドレスとユーザ名を入れるレコードを登録する様にしている．このレコードには，テーブルに対応するアプリケーションの使用許可が出されているユーザの情報が登録される．このテーブルを用いて最大同時接続数による制御の使用許可・不許可の判定を行う．

制御情報 DB の実装は MySQL を用いている．

2.3 従来システムの問題点

従来システムにおいて制御対象にできるアプリケーションは，事前に管理者が制御情報 DB に登録しているもののみであった．前述したように制御対象アプリケーションを制御情報 DB に登録する際にはアプリケーションのハッシュ値，ファイルサイズなどが必要であり，あらかじめ学生 PC 上の全てのアプリケーションを制御対象として登録しておくことは管理者の作業負担を考慮すると現実的に難しい．そのため，運用の際には事前に教員から授業で必要となるアプリケーションを管理者に申請してもらい，それを制御対象として登録しておくという方法を用いなければならない．しかしながら，この方法においては依然として管理者の負担が大きく，教員の申請の負担も増えてしまい，教員からの申請忘れなどにより実際の授業の際に必要となるアプリケーションが制御対象にないといった場合が生じる可能性がある．このような場合，授業中に教員が管理者に連絡を入れて必要なアプリケーションを制御対象として管理者が登録しなければならない．

ところが，授業中などのシステムの起動中に制御対象アプリケーションを登録しようとすると新たな問題が生じる．従来システムにおいて教員・学生それぞれのプログラムの制御対象アプリケーションリストはプログラム起動時に制御情報 DB から取得後は固定であった．そのため，システム起動中での制御対象アプリケーションの登録時には教員・学生それぞれのプログラムを再起動する必要があり，オンデマンドな制御が行えなかった．また，教員が USB メモリなどにアプリケーションのデータを入れて教室に持ち込む場合もある．このようなアプリケーションは，授業中に教員から管理者が連絡を受けたとしてもアプリケーションのデータが無いため制御対象として制御情報 DB に登録することができなかった．

さらに，従来の実装ではデフォルト禁止への切り替え時にすでに起動中のアプリケーションを強制終了する機能が実装できていなかったため，教員があるユーザをデフォルト禁止状態に変更したとしてもその学生がすでに起動していたアプリケーションを強制終了することができなかった．この問題の解決案として，教員がひとつひとつのアプリケーションを個別に指定して禁止とし，強制終了させた後にデフォルト禁止に切り替えるという方法をとることは可能だが，この方法は手間がかかり現実的ではない上に，学生が制御対象アプリケーションリストにないアプリケーションを起動していた場合，それを強制終了することができない．そのため，学生はそのままそのアプリケーションを使用できてしまう．

これらの問題点を解決するために本論文で提案する

機能を次章以降で示す.

3. 教員による講義中での制御対象追加機能

3.1 実 現 方 針

まず，管理者しか制御対象アプリケーションを登録できず，教員が望んだときにすぐ制御対象アプリケーションを追加できないという問題点を解決するために教員自身が制御対象アプリケーションをその場で制御情報 DB に登録できる仕組みを構築する．この機能を実現すれば，管理者が即時対応で制御対象アプリケーションを登録する負担を減らすこともできる．

また，プログラム起動中に制御対象アプリケーションを登録しても反映されないという問題点を解決するために DB に制御対象アプリケーションの登録が完了した後に，教員用制御情報変更プログラムおよびアプリケーション実行制御プログラム内の制御対象アプリケーションリストにその登録結果が自動的に反映される様に新たな機能を追加する．

3.2 設　　計

教員が制御対象アプリケーションを制御情報 DB に登録できる様にするにあたり，制御情報 DB への登録の際には前述したように登録するアプリケーションの名前とハッシュ値とファイルサイズと key を登録する必要がある．これらの値を教員に直接入力させるのは現実的ではないため，ハッシュ値とファイルサイズは教員のプログラム上で，登録するアプリケーションの実行ファイルを選択することで自動で計算し，それを送信するようにする．また，サーバプログラムがその情報を受信すると乱数を用いて key を生成し，これらの情報を合わせて制御情報 DB に制御対象アプリケーションの登録を行うようにする．そして，各プログラムの制御対象アプリケーションリストに登録結果を反映させるために制御対象アプリケーションが追加されたという通知と追加されたアプリケーションに関する情報をサーバプログラムからその他のプログラムに対して送信する．その後，送信された情報を教員・学生それぞれのプログラムが受け取るとプログラム中の制御対象アプリケーションリストを受信した情報を元に更新するようにする．

3.3 実　　装

前節の設計に基づいて，教員が制御対象に登録するアプリケーションのハッシュ値・ファイルサイズの計算および計算結果の送信を行うインターフェースを作成した．このインターフェースを図 3 に示す.

図 3 では制御対象に登録するアプリケーションとして acrobat reader を選択した状態を示している．このインターフェースは教員用制御情報変更プログラム上に作成されており，上の「実行ファイルを選択」ボタンを押すと制御対象アプリケーションに登録するアプリケーションの exe ファイルを選択することができる．exe ファイルを選択すると，プログラムによって自動的に図 3 のようにハッシュ値とファイルサイズが計算されて表示される．そして，この状態で下の「アプリケーションを制御対象に登録」ボタンを押すことで計算された情報にアプリケーション名を加えた情報がサーバへ送信される．

図 3 制御対象登録インターフェース
Fig. 3 Interface of registration for control.

続いて，教員からの制御対象アプリケーションの追加の要求があった場合，DB に制御対象アプリケーションを登録するようにサーバプログラムを変更した．まず，要求を受け取ると，サーバプログラムは DB の制御対象アプリケーションテーブルに既に登録されているアプリケーションのハッシュ値と受け取った情報の中のハッシュ値を比較する．こうすることで，制御対象として追加の要求があったアプリケーションがすでに制御対象アプリケーションとして登録されているかどうかを判定し，制御対象として登録されていない場合のみ制御対象として制御情報 DB に登録を行うようにした．判定の結果，すでに制御対象アプリケーションとして登録されていると判定された場合は，制御情報変更プログラムに対してその旨のメッセージを送り，制御情報変更プログラム上ですでにアプリケーションが制御対象として登録されていて制御対象に登録できない旨のエラーメッセージを表示するようにした．また，制御対象アプリケーションとして登録されていない場合，制御情報 DB への制御対象アプリケーションの登録のステップへと進む.

二重登録防止の判定にアプリケーション名でなくハッシュ値を用いているのは，アプリケーションの更新などでアプリケーション名はそのままでハッシュ値のみが変更される場合に，古いバージョンのアプリケーションが同一のアプリケーションと判定され，登録が行えないためである．もしもこのようなことが起こると，ハッシュ値の変更により，以前まで制御情報 DB に登録されていた情報では制御ができなくなっていても DB に制御対象アプリケーションとして更新後

のアプリケーションの情報を新たに制御対象として登録することができなくなってしまい，更新されたアプリケーションの制御ができないままになってしまう．

また，制御対象アプリケーションを登録する際には，前述したように key というレジストリに書き込む際の識別子が必要となるため，サーバプログラム内で乱数を用いて生成する．この key という値は複数の乱数をつないだ数字の列であり，全ての桁が完全に同じでなければレジストリ書き込みの動作には問題は生じないため，全ての桁が完全に同じ key をもつものが制御対象アプリケーションテーブルに登録されていない場合は受け取ったアプリケーション名，ハッシュ値，ファイルサイズ，key を制御対象アプリケーションテーブルに書き込む．もし，完全に同一の key をもつものが存在すればもう一度 key の生成をやり直す．

制御対象アプリケーションテーブルへの登録が完了すると，後に制御情報を登録する際に必要となるため，制御対象として登録したアプリケーションに関する教員・管理者の制御情報テーブルとアプリケーション使用許可中のユーザテーブルをサーバプログラムによって SQL 文を用いることによって自動生成し，制御情報変更プログラムとアプリケーション実行制御プログラムに対して制御対象アプリケーションが追加された旨と追加されたアプリケーションを含む制御対象アプリケーションのリストを送信する．そして，送信されたリストを制御情報変更プログラムとアプリケーション実行制御プログラムが受信するとそれぞれのプログラム中の制御対象アプリケーションリストを受信したリストを元に更新することで，新たに登録された制御対象アプリケーションをリストの中に反映させる様にした．

4. デフォルト禁止状態へ変更時の起動中アプリケーション強制終了機能

4.1 実 現 方 針

前述したように個別のアプリケーションの強制終了は起動中のプロセスを検出し，そのプロセスの中から該当のプロセスを判定して，一定時間後に強制終了をかけることで実現している．原則全てのアプリケーションを禁止とするデフォルト禁止への変更時に学生 PC 上で起動しているプロセス全てに強制終了をかければよいと考えたが，この方法では問題が生じる．ただ単にユーザの状態がデフォルト禁止状態に変わったときに学生 PC 上の全てのアプリケーションを強制終了させようとして学生の PC 上で起動しているプロセス全てに対して強制終了をかけると，windows explorer や実行制御プログラム自身などのプロセスやアプリケーションのものではないプロセスに対してまでも強制終了をかけてしまい，ユーザの PC の動作やシステム全体の動作に悪影響を及ぼしてしまう．最悪の場合，デフォルト禁止状態でアプリケーション実行制御プログラムが停止し，全てのアプリケーションの起動ができない状態で固まってしまう可能性もある．そのため，強制終了対象とするプロセスを最低限強制終了をしてはいけないもの以外全てのアプリケーションのものに限定する必要がある．

4.2 設　　計

強制終了対象を限定するに当たり，まずは学生が起動したアプリケーションのものではないようなバックグラウンドで動いているプロセスを強制終了対象から外すことにする．この判定を実現するにあたり，学生が起動したアプリケーションはすべてメインウインドウをもつと考えメインウインドウをもたないプロセスを全て強制終了対象から外すようにする．しかし，この条件のみでは windows explorer やアプリケーション実行制御プログラム本体のようなメインウインドウを持つプロセスは強制終了の対象となってしまう．そのため，制御情報 DB に強制終了をさせてはいけないアプリケーションのホワイトリストを作成し，デフォルト禁止状態への変更命令と一緒にそれを送信するようにして受信したアプリケーション実行制御プログラムにおいて強制終了対象から除外する．これらの条件のいずれも満たさないプロセスを強制終了をかけてもよいアプリケーションのプロセスであると判定して，作業内容の保存のために一定時間の猶予を持たせた後，強制終了をかけることとする．ホワイトリストの登録は管理者が行うようにし，教員やユーザは行えないようにする．

4.3 実　　装

前節の方針に基づいてこの機能を作成するにあたり，まずは管理者がホワイトリストを登録するためのインターフェースを作成した．そのインターフェースを図 4 に示す．

図 4　ホワイトリスト登録インターフェース
Fig. 4　Whitelist registration interface.

このインターフェースは管理者が自身の管理者名と

パスワードを用いてログインして使用する．右側のリストには現在のホワイトリストに登録されている内容をログイン時に取得して表示する．そして，実行ファイルを選択というボタンを押すとホワイトリストに登録するアプリケーションの exe ファイルを選択することができ，選択するとアプリケーションの exe ファイルからハッシュ値を計算し，表示する．図 4 の例では windows explorer を選択している．アプリケーションを選択した状態で，下のふたつのボタンを押すとホワイトリストの登録・削除の要求と計算結果をサーバへ送信する．

サーバ上の制御情報 DB にはこの送信されたホワイトリストの情報を格納するためのホワイトリストテーブルと管理者ログイン認証用の管理者情報テーブルを新たに作成した．ホワイトリストテーブルにはホワイトリストとして登録されたアプリケーションの名前とハッシュ値を登録し，管理者情報テーブルには管理者の管理者名とログイン用パスワードを登録する．

ホワイトリスト登録インターフェースからホワイトリストの登録の要求を受け取るとサーバプログラムではホワイトリストテーブル内の全レコードのハッシュ値と受け取ったハッシュ値とを比較し，すでにホワイトリストに登録されているかどうかをチェックする．そして，ホワイトリストテーブルにないと判断されればホワイトリストテーブルへの登録が行われた後，ホワイトリスト登録インターフェースにメッセージが返されてホワイトリスト登録インターフェース上に登録成功メッセージを表示する．また，すでにホワイトリストテーブルにあると判断された場合はその旨のメッセージが返され，登録失敗メッセージを表示する．この登録されたホワイトリストテーブルのアプリケーションのハッシュ値は，ユーザをデフォルト禁止状態へと変更する制御情報とともにサーバからユーザのもとへ送信し，デフォルト禁止状態への変更時に強制終了するプロセスの判断基準に用いる．

アプリケーション実行制御プログラムがデフォルト禁止状態への変更の制御情報と同時にホワイトリスト内のハッシュ値の情報を受け取ると，始めに GetProcess メソッドを用いてユーザの PC 上で起動している全てのプロセスの情報を取得する[8]．その後，メインウインドウを持たないプロセスを強制終了対象から外すため，MainWindowHandle プロパティの値が IntPtr．Zero(=未設定であり, メインウインドウを持たない) でなければ，それらプロセスに対してのみ次の判定を行うようにし，それ以外のプロセスを強制終了対象から除外する．そして，ここでメインウインドウをもつとみなされたプロセスのパスからハッシュ値を計算し，サーバから受け取ったホワイトリストに登録されているアプリケーションのハッシュ値と比較をする．その結果，ホワイトリストの中に存在しないと判定されたプロセスに対してのみ強制終了の処理を行い，ホワイトリスト内のアプリケーションに対しては強制終了の処理を行わない．強制終了の方法については個別のアプリケーションに対する強制終了の方法と同様に，一定時間強制終了まで猶予を持たせた後，該当のプロセスに対して kill 命令を実行する．

5. 動作確認実験

今回実装した二つの機能の動作を確認するために動作確認実験を行った．

[実験内容]

アプリケーションを新たに制御対象として登録を行った後登録結果が制御情報変更プログラムおよびアプリケーション実行制御プログラム内の制御対象アプリケーションリストに反映されることを確認する．その後，新たに制御対象に加えたアプリケーションに対して制御を行い，制御が正しく行えることを確認する．その後に，ホワイトリストにアプリケーションを登録し，アプリケーション実行制御プログラムの状態をデフォルト禁止に変化させ，ホワイトリストに登録されているアプリケーション以外の起動中アプリケーションが一定時間後に強制終了されることを確認する．

[実験開始前の設定]

実験開始前には制御対象アプリケーションとして Internet Explorer と Acrobat Reader を登録し，ホワイトリスト内にアプリケーション実行制御プログラムの実行ファイルのみを登録した．ユーザ PC 上では Microsoft Word と Internet explorer と Acrobat Reader と thunderbird を起動しておいた．

[実験手順]

(1) 制御情報変更プログラム中の制御対象登録インターフェースから Microsoft Word を制御対象アプリケーションとして登録する．
(2) 制御情報変更プログラムとアプリケーション実行制御プログラム中の制御対象アプリケーションリストが更新されることを確認する．
(3) 新たに登録した Microsoft Word の起動を禁止にして，Microsoft Word が強制終了されて，起動が制御されることを確認する．
(4) ホワイトリスト登録インターフェースを用いて，ホワイトリストに windows explorer を登録する．
(5) アプリケーション実行制御プログラムをデフォルト禁止状態に変更し，ホワイトリストに登録されていない Internet explorer と Acrobat Reader と thunderbird が強制終了されることを確認する．

[実験結果] まず，制御対象登録インターフェースから Microsoft Word を制御対象として登録したところ制御対象として登録が成功した旨のメッセージが表示された．その後，制御情報変更プログラムおよびアプ

リケーション実行制御プログラム中の制御対象アプリケーションリストを確認したところ，新たに制御対象として登録した Micorsoft Word がリストに加わっていることを確認し，制御対象アプリケーションリストの更新が正常に実行されたことを確認した．

続いて，制御情報変更プログラムを用いて，アプリケーション実行制御プログラムに対して新たに制御対象アプリケーションとして登録した Microsoft Word の使用を禁止とするよう制御情報の変更を行った．その結果，起動中の Microsoft Word は強制終了され，強制終了後にもう一度起動しようとしても起動をすることができないことを確認した．

最後に，ホワイトリスト登録インターフェースを用いてホワイトリストに windows explorer を登録した後に，アプリケーション実行制御プログラムの状態をデフォルト禁止状態に変更した．このとき強制終了されたアプリケーションは Internet explorer と Acrobat Reader と thunderbird のみであり，あらかじめホワイトリストに登録されていたアプリケーション実行制御プログラム本体，新たにホワイトリストに登録した windows explorer，アプリケーション以外のプロセスが強制終了対象とならず，そのほかのアプリケーションのプロセスのみが作業の保存のための一定時間の猶予後に強制終了されることを確認した．

6. むすび

本論文では，アプリケーション実行制御システムにおいて教員自身が制御対象アプリケーションを登録する機能およびユーザの状態をデフォルト禁止状態へ切り替えられたときに起動中のアプリケーションを強制終了させる機能を提案し，設計と実装および動作確認を行った．

従来のアプリケーション実行制御システムの実装では，管理者以外が制御対象アプリケーションを登録する仕組みが存在せず，講義中などのプログラムの起動中に管理者が制御対象アプリケーションの登録を行ったとしても，登録された制御対象アプリケーションを制御情報変更プログラムおよびアプリケーション実行制御プログラム内の制御対象アプリケーションリストに反映させる仕組みも存在しなかった．そのため，講義中などにプログラムを実行したままで制御対象アプリケーションを増やすことは不可能で，管理者の制御対象アプリケーション登録のたびに全てのプログラムを再起動する必要があった．

また，デフォルト禁止への切り替え時に起動中のアプリケーションを強制終了させる機能が実装できていなかったため，ユーザの状態ががデフォルト禁止状態に変更されても，すでにユーザが起動しているアプリケーションに関しては継続して使用可能であった．しかし，今回の研究で提案した機能を実装したことによってこれらの問題を解決することができた．

今後の課題としては，多数のユーザに対する制御内容変更メッセージ送信の負荷のテスト，教員用プログラムにおいて前回のログイン時に設定した制御情報を再利用する機能の検討，残りライセンス数が少ない場合に，関連授業の履修者に対して優先的に利用許可を出す利用可否判定手法の検討，授業の履修情報や所属などのユーザ情報に応じた制御対象ユーザの指定方法の検討などが挙げられる．

参考文献

1) 文部科学省,“ 教育の情報化に関する手引き，”http://www.mext.go.jp/amanu/shotou/zyouhou/1259413.htm.
2) 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.
3) 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.
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) 川上崇，河野圭太，山井成良,“ 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，”インターネットと運用技術シンポジウム 2010 論文集，Vol.2010，pp1-8，Dec．2010
6) WindowsXP/2000 のレジストリ，http://www.dr-pc.jp/reg2.htm.
7) 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.
8) MSDN ライブラリ,“ Windows プロセスの監視と管理，”http://msdn.microsoft.com/ja-jp/library/4z4t818a(v=vs.80).aspx)