

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-67156

(P2010-67156A)

(43) 公開日 平成22年3月25日(2010.3.25)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 9/54 (2006.01)	G06F 9/46 480F	5B046
G06F 17/50 (2006.01)	G06F 17/50 614A	
	G06F 17/50 601A	

審査請求 未請求 請求項の数 6 O L (全 12 頁)

(21) 出願番号	特願2008-234840 (P2008-234840)	(71) 出願人	000001052 株式会社クボタ 大阪府大阪市浪速区敷津東一丁目2番47号
(22) 出願日	平成20年9月12日 (2008.9.12)	(74) 代理人	100104444 弁理士 上羽 秀敏
		(74) 代理人	100123906 弁理士 竹添 忠
		(72) 発明者	堀 武雄 大阪府八尾市神武町2番35号 株式会社クボタ久宝寺事業センター内
		Fターム(参考)	5B046 CA06 KA05

(54) 【発明の名称】 クライアントサーバシステム

(57) 【要約】

【課題】CADが更新されて設計データの様式が新しくなっても、サーバもクライアントも改変することなく、その新しい様式の設計データをサーバの既存APIで処理することができるクライアントサーバ型CADデータ管理システムを提供する。

【解決手段】このシステム10は、サーバ12とクライアント14との間にプロキシ16を介在させ、サーバ12のAPI1~API nをフックするアドインAI1~AI mをプロキシ16に追加し、アドインAI1~AI mがAPI1~API nを呼び出すと、引数を変更したり、戻り値を変更したりする。アドインAI1~AI mがフックするAPI1~API nは、API・アドインマップ26に登録される。

【選択図】 図1

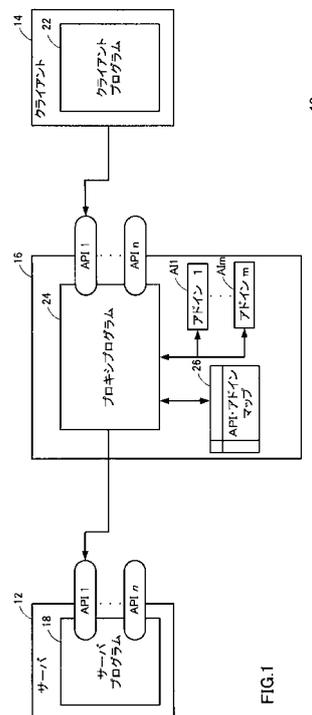


FIG. 1

【特許請求の範囲】**【請求項 1】**

サーバと、クライアントと、前記サーバ及び前記クライアント間に接続されるプロキシとを備えるクライアントサーバシステムであって、

前記サーバは、複数の A P I を含み、

前記クライアントは、前記複数の A P I の中から所望の A P I を呼び出す A P I 呼出手段を含み、

前記プロキシは、

前記複数の A P I に対応し、各々が、対応する A P I をフックし、かつ、対応する A P I を呼び出して引数及び / 又は戻り値を変更する複数のアドインと、

前記複数のアドインの中から前記 A P I 呼出手段により呼び出された A P I に対応するアドインを特定するアドイン特定手段と、

前記アドイン特定手段により特定されたアドインを呼び出すアドイン呼出手段とを含む、クライアントサーバシステム。

【請求項 2】

請求項 1 に記載のクライアントサーバシステムであって、

前記プロキシはさらに、

前記 A P I をフックしているアドインを登録した A P I ・アドインマップと、

前記 A P I 呼出手段により呼び出された A P I をフックしているアドインが前記 A P I ・アドインマップに登録されているか否かを判断する登録判断手段と、

前記登録判断手段による判断の結果、当該アドインが登録されていない場合、前記 A P I 呼出手段により呼び出された A P I を呼び出す A P I 呼出手段とを含み、

前記登録判断手段による判断の結果、当該アドインが登録されている場合、前記アドイン特定手段は当該アドインを特定する、クライアントサーバシステム。

【請求項 3】

請求項 2 に記載のクライアントサーバシステムであって、

前記プロキシはさらに、

前記アドインがフック可能な A P I を前記アドインに問い合わせる前記 A P I ・アドインマップを作成するマップ作成手段を含む、クライアントサーバシステム。

【請求項 4】

サーバ及びクライアント間に接続されるプロキシであって、

前記サーバに搭載される複数の A P I に対応し、各々が、対応する A P I をフックし、かつ、対応する A P I を呼び出して引数及び / 又は戻り値を変更する複数のアドインと、

前記複数のアドインの中から前記クライアントから呼び出された A P I に対応するアドインを特定するアドイン特定手段と、

前記アドイン特定手段により特定されたアドインを呼び出すアドイン呼出手段とを備える、プロキシ。

【請求項 5】

請求項 4 に記載のプロキシであってさらに、

前記 A P I をフックしているアドインを登録した A P I ・アドインマップと、

前記 A P I 呼出手段により呼び出された A P I をフックしているアドインが前記 A P I ・アドインマップに登録されているか否かを判断する登録判断手段と、

前記登録判断手段による判断の結果、当該アドインが登録されていない場合、前記 A P I 呼出手段により呼び出された A P I を呼び出す A P I 呼出手段とを備え、

前記登録判断手段による判断の結果、当該アドインが登録されている場合、前記アドイン特定手段は当該アドインを特定する、プロキシ。

【請求項 6】

請求項 5 に記載のプロキシであってさらに、

前記アドインが可能な A P I を前記アドインに問い合わせる前記 A P I ・アドインマップを作成するマップ作成手段を備える、プロキシ。

10

20

30

40

50

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、クライアントサーバシステムに関し、さらに詳しくは、C A D (Computer Aided Design)用の設計データを管理するクライアントサーバ型C A Dデータ管理システムに関する。

【背景技術】**【0002】**

C A D (アプリケーションプログラム)で作成された設計データの管理には、C A Dとは別に、専用のデータ管理プログラムが用いられる。このデータ管理プログラムは、作成された設計データを保存したり、検索したり、参照したり、再利用したりする機能を有し、チームで1つの製品を設計するためには必須のソフトウェア(以下、「管理ソフト」という。)である。この管理ソフトはC A Dベンダから提供されているが、C A Dベンダ以外のベンダからも提供されている。

10

【0003】

データ管理プログラムを用いたクライアントサーバ型C A Dデータ管理システムは、設計者がC A Dを操作して設計データを作成するクライアントコンピュータ(以下、単に「クライアント」という。)と、設計データを一元的に管理するサーバコンピュータ(以下、単に「サーバ」という。)とを備える。このシステムにおいては、サーバが各種A P I (Application Programming Interface)を備え、クライアントがその中から所望のA P Iを呼び出す。したがって、C A Dが更新されれば、これらのA P Iも更新されなければならない。

20

【0004】

しかしながら、データ管理プログラムの複雑化に伴い、C A Dベンダによる更新の対応が遅れているのが現状である。また、A P Iを追加できるように設計されていないデータ管理プログラムの場合、C A DベンダしかA P Iを追加することができない。したがって、C A Dベンダが対応しなければ、管理ソフトベンダはA P Iを追加することができない。また、A P Iを追加できるように設計されているデータ管理プログラムの場合であっても、A P Iを追加すると、C A Dやデータ管理プログラムが更新されるたびに、追加したA P Iのメンテナンスをしなければならない。また、A P I自体を改変することは更新されたC A Dを熟知しているC A Dベンダにとっては容易であるが、更新されたC A Dを熟知していない管理ソフトベンダにとっては困難である。

30

【0005】

なお、特開2000-222324号公報(特許文献1)には、テキスト、イメージ、プログラム、ハイパーリンク、表、アプレットなど、H T M Lで記述できる全ての種類のオブジェクトを追加できるウェブページカスタマイズシステムが開示されている。このシステムは、ユーザのコンピュータから、カスタマイズしたいページをプロキシサーバを介して要求し、プロキシサーバが、要求されたページにカスタマイズのためのプログラムを埋め込み、コンピュータのブラウザにより、該プログラムの埋め込まれたページを表示し、該プログラムが、カスタマイズ操作のための制御パネルを表示し、制御パネルを用いたユーザのカスタマイズ操作に応じて、ページをカスタマイズするように構成されている。しかしながら、同公報には、後述する本発明の課題及びその解決手段は何ら開示も示唆もされていない。

40

【特許文献1】特開2000-222324号公報

【発明の開示】**【発明が解決しようとする課題】****【0006】**

本発明の目的は、あるアプリケーションプログラムが更新され、そのプログラムで作成されるデータの様式が新しくなっても、サーバもクライアントも改変することなく、その新しい様式のデータをサーバの既存A P Iで処理することができるクライアントサーバシ

50

ステムを提供することである。

【課題を解決するための手段及び発明の効果】

【0007】

本発明によるクライアントサーバシステムは、サーバと、クライアントと、サーバ及びクライアント間に接続されるプロキシコンピュータ（以下、単に「プロキシ」という。）とを備える。サーバは、複数のAPIを含む。クライアントは、複数のAPIの中から所望のAPIを呼び出すAPI呼出手段を含む。プロキシは、複数のアドインと、アドイン特定手段と、アドイン呼出手段とを含む。複数のアドインは、複数のAPIに対応する。各アドインは、対応するAPIをフックし、かつ、対応するAPIを呼び出して引数及び/又は戻り値を変更する。アドイン特定手段は、複数のアドインの中からAPI呼出手段により呼び出されたAPIに対応するアドインを特定する。アドイン呼出手段は、アドイン特定手段により特定されたアドインを呼び出す。

10

【0008】

本発明によれば、サーバ及びクライアント間にプロキシを接続し、サーバのAPIをフックするアドインをプロキシに追加し、アドインがAPIを呼び出して引数及び/又は戻り値を変更するようにしているため、処理対象となるデータの様式が更新されても、クライアント及びサーバを改変することなく、その新しい様式のデータを既存のAPIで処理することができる。

【0009】

好ましくは、プロキシはさらに、API・アドインマップと、登録判断手段と、API呼出手段とを含む。API・アドインマップには、APIをフックしているアドインが登録される。登録判断手段は、API呼出手段により呼び出されたAPIをフックしているアドインがAPI・アドインマップに登録されているか否かを判断する。API呼出手段は、登録判断手段による判断の結果、当該アドインが登録されていない場合、API呼出手段により呼び出されたAPIを呼び出す。一方、登録判断手段による判断の結果、当該アドインが登録されている場合、アドイン特定手段は当該アドインを特定する。

20

【0010】

この場合、アドインがサーバのAPIをフックしていれば、プロキシがアドインを呼び出す。アドインがサーバのAPIをフックしていなければ、クライアントがプロキシをスルーしてサーバのAPIを呼び出す。

30

【0011】

好ましくは、プロキシはさらに、アドインがフック可能なAPIをアドインに問い合わせるAPI・アドインマップを作成するマップ作成手段を含む。

【0012】

この場合、API・アドインマップは自動的に作成される。

【発明を実施するための最良の形態】

【0013】

以下、図面を参照し、本発明の実施の形態を詳しく説明する。図中同一又は相当部分には同一符号を付してその説明は繰り返さない。

【0014】

図1を参照して、本発明の実施の形態によるクライアントサーバ型CADデータ管理システム10は、サーバ12と、1又は2以上のクライアント14と、プロキシ16とを備える。プロキシ16は、サーバ12及びクライアント14の間に接続され、サーバ12に対してクライアントとして機能し、クライアント14に対してサーバとして機能する。

40

【0015】

サーバ12には、CADデータ管理用のサーバプログラム18がインストールされ、複数のAPI1～API_nが搭載される。各クライアント14には、CADデータ管理用のクライアントプログラム22がインストールされる。サーバプログラム18、API1～API_n及びクライアントプログラム22は、CADソフトベンダから提供されている既存のもので、設計データの保存、検索、参照、再利用などの基本機能をサーバ12及びク

50

クライアント 14 に実現させるためのものである。

【0016】

プロキシ 16 には、プロキシプログラム 24 がインストールされ、さらに複数のアドインプログラム（以下、単に「アドイン」という。）A I 1 ~ A I m が追加インストールされる。アドイン A I 1 ~ A I m は、サーバ 12 の A P I 1 ~ A P I n をフックし、上記基本機能と異なる各種拡張機能をプロキシ 16 に実現させるためのもので、必要に応じて追加、削除、有効化、無効化などが可能である。プロキシ 16 にはまた、プロキシプログラム 24 によって作成される A P I ・アドインマップ 26 が記憶される。

【0017】

なお、サーバ 12、クライアント 14 及びプロキシ 16 は全てコンピュータであるから、図示は省略するが、C P U (Central Processing Unit)、メモリ、ハードディスク、各種入出力装置などを備える。サーバプログラム 18、クライアントプログラム 22、A P I 1 ~ A P I n、アドイン A I 1 ~ A I m、及び A P I ・アドインマップ 26 は、各コンピュータのハードディスク又はメモリに記憶される。

【0018】

以下、C A D データ管理システム 10 の動作を説明する。

[初期化処理]

まず図 2 を参照して、初期化処理を説明する。

【0019】

プロキシ 16 はプロキシプログラム 24 に従ってステップ S 11 ~ S 16 を実行し、アドイン A I 1 に従ってステップ S 21 ~ S 22 を実行し、アドイン A I m に従ってステップ S 23 ~ S 24 を実行する。その他のアドインも同様である。以下、より具体的に説明する。

【0020】

プロキシプログラム 24 は、あらかじめ作成されている初期設定ファイル 28 を読み込み、以下のステップで呼び出すべきアドインを決定する (S 11)。初期設定ファイル 28 には、プロキシ 16 に追加インストールされている全てのアドイン A I 1 ~ A I m の識別情報が優先順位とともにあらかじめ登録されている。

【0021】

プロキシプログラム 24 は、決定したアドイン A I 1 ~ A I m のうち最初のアドイン A I 1 を呼び出す (S 12)。この呼出に応じて、アドイン A I 1 は起動する (S 21)。プロキシプログラム 24 は、起動したアドイン A I 1 に対し、アドイン A I 1 がフック可能な A P I を問い合わせる (S 13)。この問合せに応じて、アドイン A I 1 は、自身がフック可能な A P I をプロキシプログラム 24 に返答する (S 22)。

ここで、一部の特殊なアドイン（たとえば後述するログアドイン）は、プロキシプログラム 24 からの問合せに応じて、逆にプロキシ 24 に実装されている全ての A P I 1 ~ A P I n を表した一覧を要求する。プロキシプログラム 24 は、この特殊なアドインからの要求に応じて、全ての A P I 1 ~ A P I n を表した一覧を提供する。この場合、特殊なアドインはサーバ 12 に追加又は変更された A P I 1 ~ A P I n を知らなくても A P I 1 ~ A P I n をフックすることができる。

【0022】

プロキシプログラム 24 は、その他のアドインについても上記と同様の処理を繰り返した後、最後のアドイン A I m を呼び出す (S 14)。この呼出に応じて、アドイン A I m は起動する (S 23)。プロキシプログラム 24 は、起動したアドイン A I m に対し、アドイン A I m がフック可能な A P I を問い合わせる (S 15)。この問合せに応じて、アドイン A I m は、自身がフック可能な A P I をプロキシプログラム 24 に返答する (S 24)。

【0023】

そして、プロキシプログラム 24 は、アドイン A I 1 ~ A I m からの返答に基づいて、A P I ・アドインマップ 26 を作成する (S 16)。作成された A P I ・アドインマップ

10

20

30

40

50

26の一例を図3に示す。API・アドインマップ26には、各アドインがフックするAPIが登録される。たとえば図3に示した例では、アドインAI1はAPI1及びAPI3をフックし、アドインAI2はAPI2のみをフックし、アドインAI3は全てのAPI1～API n をフックし、アドインAI m はAPI $n-1$ をフックしている。各アドインは、1つだけでなく複数のAPIをフックする場合がある。また、複数のアドインが同じAPIをフックする場合もある。

【0024】

この初期化処理によれば、後述するプロキシ16の介入処理に必要なAPI・アドインマップ26があらかじめ自動的に作成される。

【0025】

[要求時の介入処理]

次に図4を参照して、クライアント14からサーバ12への要求時におけるプロキシ16の介入処理を説明する。

【0026】

クライアント14は、クライアントプログラム22に従ってステップS31～S33を実行する。プロキシ16は、プロキシプログラム24に従ってステップS41～S50を実行し、かつ、アドインプログラムAI1～AI m に従ってステップS61を実行する。サーバ12は、サーバプログラム18に従ってステップS71～S74を実行する。以下、より具体的に説明する。

【0027】

クライアント14は、サーバ12のAPI x ($x=1\sim n$)を呼び出すために、コールコマンドをサーバ12に向けて送信する(S31)。サーバ12とクライアント14の間にはプロキシ16が介在しているため、プロキシ16はクライアント14からサーバ12への要求を遮ってコールコマンドを受信する(S41)。プロキシプログラム24は、受信したコールコマンドを解析し、クライアント14が呼び出そうとしているAPI x に対応する1又は2以上のアドインAI y ($y=1\sim m$)がAPI・アドインマップ26に登録されているか否かを判断する(S42)。

【0028】

対応するアドインAI y が登録されていない場合(S42でNO)、プロキシプログラム24はそのコールコマンドをそのまま通過させることにより、サーバ12のAPI x を呼び出す(S43)。サーバ12はそのコールコマンドを受信し(S71)、API x の実行を開始する(S72)。

【0029】

一方、対応する1又は2以上のアドインAI y が登録されている場合(S42でYES)、プロキシプログラム24は、そのアドインAI y を特定し(S44)、そのアドインAI y がフックしているAPI x を正規化し、引数と戻り値のインターフェイスを所定の書式に合わせる(S45)。

【0030】

続いて、プロキシプログラム24は、特定したアドインAI y を順番に呼び出し(S46, S48)、これにより要求された処理をアドインAI y に移譲する。この呼出に応じて、アドインAI y は、特有の拡張機能を実行する他、コールコマンドに含まれる引数を変更した上でAPI x に引き渡したり、API x から引き渡された戻り値を変更した上でプロキシプログラム24に返却したりする(S61, S63)。アドインAI1～AI m の中には、特有の拡張機能を実行する他、引数のみを変更するもの、戻り値のみを変更するもの、引数及び戻り値の両方を変更するもの、あるいは戻り値を独自に作成するものがある。ここで、戻り値を独自に作成するとは、アドインAI y がクライアント14からの呼出内容に関係なく所定の戻り値を作成することであり、戻り値を変更することの一種である。そして、アドインAI y は、API x の呼出を中止するか否かを判断する(S62, S64)。

【0031】

10

20

30

40

50

アドイン A I y が A P I x の呼出を中止すると判断した場合 (S 6 2 , S 6 4 で Y E S)、プロキシプログラム 2 4 は、アドイン A I y が返した戻り値をクライアント 1 4 に返却する (S 4 7 , S 4 9)。クライアント 1 4 は、プロキシ 1 6 から返却された戻り値を受信し、A P I x の呼出を完了する (S 3 2 , S 3 2)。

【 0 0 3 2 】

一方、ステップ S 4 4 で特定された 1 又は 2 以上のアドイン A I y の全てが A P I x の呼出を中止すると判断した場合 (S 6 2 , S 6 4 で N O)、プロキシプログラム 2 4 はクライアント 1 4 からの呼出をそのまま通過させ、原則通りサーバ 1 2 の A P I x を呼び出す (S 5 0)。サーバ 1 2 はこの呼出を受信し (S 7 3)、A P I x の実行を開始する (S 7 4)。

10

【 0 0 3 3 】

[応答時の介入処理]

次に図 5 を参照して、サーバ 1 2 からクライアント 1 4 への応答時におけるプロキシ 1 6 の介入処理を説明する。

【 0 0 3 4 】

クライアント 1 4 は、クライアントプログラム 2 2 に従ってステップ S 3 4 ~ S 3 5 を実行する。プロキシ 1 6 は、プロキシプログラム 2 4 に従ってステップ S 5 1 ~ S 5 8 を実行し、かつ、アドインプログラム A I 1 ~ A I m に従ってステップ S 6 5 ~ S 6 6 を実行する。サーバ 1 2 は、サーバプログラム 1 8 に従ってステップ S 7 5 を実行する。以下、より具体的に説明する。

20

【 0 0 3 5 】

サーバ 1 2 は A P I x の実行を開始し (S 7 2 , S 7 4)、完了すると、その戻り値をクライアント 1 4 に向けて送信する (S 7 5)。しかし、サーバ 1 2 とクライアント 1 4 との間にはプロキシ 1 6 が介在しているため、プロキシ 1 6 はサーバ 1 2 からクライアント 1 4 への応答を遮って戻り値を受信する (S 5 1)。

【 0 0 3 6 】

プロキシプログラム 2 4 は、A P I x に対応するアドイン A I y (y = 1 ~ m) が A P I ・アドインマップ 2 6 に登録されているか否かを判断する (S 5 2)。この判断はステップ S 4 2 の判断と同じである。

【 0 0 3 7 】

対応するアドイン A I y が登録されていない場合 (S 5 2 で N O)、プロキシプログラム 2 4 はその戻り値をそのまま通過させてクライアント 1 4 に返却する (S 5 3)。クライアント 1 4 はその戻り値を受信し、A P I x の実行を完了する (S 3 4)。

30

【 0 0 3 8 】

一方、対応するアドイン A I y が登録されている場合 (S 5 2 で Y E S)、プロキシプログラム 2 4 は、そのアドイン A I y を特定し (S 5 4)、そのアドイン A I y がフックしている A P I x を正規化する (S 5 2)。

【 0 0 3 9 】

続いて、プロキシプログラム 2 4 は、特定した 1 又は 2 以上のアドイン A I y を順番に呼び出し、これにより要求された処理をアドイン A I y に移譲する (S 5 6 , S 5 7)。この呼出に応じて、アドイン A I y は、特有の拡張機能を実行する他、戻り値を変更した上でプロキシプログラム 2 4 に返却する (S 6 5 , S 6 6)。

40

【 0 0 4 0 】

プロキシプログラム 2 4 は、アドイン A I y から返却された戻り値をクライアント 1 4 に返却する (S 5 8)。クライアント 1 4 は、プロキシ 1 6 から返却された戻り値を受信し、A P I x の呼出を完了する (S 3 5)。

【 0 0 4 1 】

[アドインの具体例]

上記アドイン A I 1 ~ A I m が有する特有の拡張機能は特に限定されないが、以下、拡張機能の具体例を 2 つ紹介する。

50

【 0 0 4 2 】

(1) セキュリティアドイン

セキュリティアドインは、ファイル削除 A P I をフックする。ファイル削除 A P I は、指定されたファイルをサーバ 1 2 から削除する特有の拡張機能を有する。ファイル削除 A P I の引数は、削除すべきファイルを特定する識別子「 I D 」である。ファイル削除 A P I の戻り値は、ファイルの削除に成功した場合は「 0 」になり、失敗した場合はそれ以外の値になる。セキュリティアドインは、クライアント 1 4 からサーバ 1 2 への要求時のみ処理を行い、サーバ 1 2 からクライアント 1 4 への応答時には処理を行わない。以下、図 6 を参照して、より具体的に説明する。

【 0 0 4 3 】

ファイル削除 A P I をフックするアドインとしてセキュリティアドインが初期設定ファイル 2 8 に登録され、図 2 に示した初期化処理により、ファイル削除 A P I をフックするアドインとしてセキュリティアドインが A P I ・アドインマップ 2 6 に登録される。

【 0 0 4 4 】

まず、クライアント 1 4 は、サーバ 1 2 のファイル削除 A P I 「DeleteFile (I D) 」の呼出を開始する (S 3 1)。プロキシプログラム 2 4 は、クライアント 1 4 が呼び出そうとしているファイル削除 A P I に対応するアドインが A P I ・アドインマップ 2 6 に登録されていると判断する (S 4 2)。

【 0 0 4 5 】

プロキシプログラム 2 4 は、ファイル削除 A P I に対応するアドインとしてセキュリティアドインを特定し (S 4 4)、ファイル削除 A P I を正規化する (S 4 5)。

【 0 0 4 6 】

続いて、プロキシプログラム 2 4 は、特定したセキュリティアドインを呼び出し、これにより要求された処理をセキュリティアドインに移譲する (S 4 6)。具体的には、ファイル削除 A P I の関数名「DeleteFile」、引数「 I D 」及び戻り値をセキュリティアドインに伝達する。この呼出に応じて、セキュリティアドインは、引数「 I D 」で特定されるファイルが削除されないように保護されているか否かを判断する (S 6 0)。

【 0 0 4 7 】

ファイルが保護されている場合 (S 6 0 で Y E S)、セキュリティアドインは、戻り値を「 1 」に変更する。戻り値「 1 」はファイルの削除に失敗したことを意味する。セキュリティアドインは、変更した戻り値「 1 」をプロキシプログラム 2 4 に引き渡し、以降のアドインが誤動作しないように A P I の呼出中止を依頼する (S 6 2)。

【 0 0 4 8 】

プロキシプログラム 2 4 は、セキュリティアドインから返却された戻り値「 1 」をクライアント 1 4 に返却する (S 4 7)。クライアント 1 4 は、プロキシ 1 6 から返却された戻り値「 1 」を受信し、ファイル削除 A P I 「DeleteFile (I D) 」の呼出を完了する (S 3 2)。

【 0 0 4 9 】

一方、ファイルが保護されていない場合 (S 6 0 で N O)、プロキシプログラム 2 4 は、サーバ 1 2 のファイル削除 A P I 「DeleteFile (I D) 」を呼び出す (S 5 0)。この呼出に応じて、サーバ 1 2 はファイル削除 A P I の実行を開始する (S 7 4)。

【 0 0 5 0 】

(2) ログアドイン

ログアドインは、サーバ 1 2 のファイルダウンロード A P I をフックする。ファイルダウンロード A P I は、指定されたファイルをサーバ 1 2 からダウンロードする機能を有する。ファイルダウンロード A P I の引数は、ダウンロードすべきファイルを特定する識別子「 I D 」である。ファイルダウンロード A P I の戻り値は、指定されたファイルのバイナリデータである。ログアドインは、サーバ 1 2 からクライアント 1 4 への応答時のみ処理を行い、クライアント 1 4 からサーバ 1 2 への要求時には処理を行わない。以下、図 7 を参照して、より具体的に説明する。

10

20

30

40

50

【 0 0 5 1 】

ファイルダウンロードAPIをフックするアドインとしてログアドインが初期設定ファイル28に登録され、図2に示した初期化処理により、ファイルダウンロードAPIをフックするアドインとしてログアドインがAPI・アドインマップ26に登録される。

【 0 0 5 2 】

サーバ12はファイルダウンロードAPI「DownloadFile(ID)」の実行を完了すると、その戻り値をクライアント14に向けて送信する(S75)。プロキシプログラム24は、サーバ12から送信された戻り値を受信し、ファイルダウンロードAPIに対応するアドインがAPI・アドインマップ26に登録されていると判断する(S52)。

【 0 0 5 3 】

プロキシプログラム24は、ファイルダウンロードAPIに対応するアドインとしてログアドインを特定し(S54)、ファイルダウンロードAPIを正規化する(S55)。

【 0 0 5 4 】

続いて、プロキシプログラム24は、特定したログアドインを呼び出し、これにより要求された処理をログアドインに移譲する(S56)。この呼出に応じて、ログアドインは、ファイルダウンロードAPIはどのクライアント14により呼び出されたのかをプロキシ16に問い合わせるために、プロキシ16に接続クライアント情報を要求する(S67)。接続クライアント情報は、ファイルダウンロードAPIを呼び出したクライアントの識別情報であって、プロキシ16で管理されている。

【 0 0 5 5 】

プロキシプログラム24は、この要求に応じて、接続クライアント情報をログアドインに返答する(S59)。ログアドインは、返答された接続クライアント情報に基づいて、ファイルのダウンロードを実行した日時、ユーザ、クライアントIP、APIなどを記録する(S68)。

【 0 0 5 6 】

プロキシプログラム24は、アドインAiyから返却された戻り値をクライアント14に返却する(S58)。クライアント14は、プロキシ16から返却された戻り値を受信し、ファイルダウンロードAPIの呼出を完了する(S35)。

【 0 0 5 7 】

以上のように、本発明の実施の形態によれば、サーバ12とクライアント14との間にプロキシ16を介在させ、サーバ12のAPI1～APInをフックするアドインAI1～AImをプロキシ16に追加し、アドインAI1～AImがAPI1～APInを呼び出し、引数を変更したり、戻り値を変更したりしているため、CADが更新されて設計データの様式が新しくなっても、サーバ12もクライアント14も改変することなく、その新しい様式の設計データを既存のAPI1～APInで処理することができる。

【 0 0 5 8 】

上記実施の形態では、サーバ12のAPI1～APInと同じAPI1～APInがプロキシ16にも実装されているが、サーバ12及びプロキシ16がAPI1～APInを共用するようにしてもよい。

【 0 0 5 9 】

また、上記実施の形態では、プロキシ16はサーバ12を構成するコンピュータとは別のコンピュータで構成されているが、プロキシプログラム24及びアドインAI1～AImをサーバ12と同じコンピュータにインストールすることにより、サーバ12及びプロキシ16を1台のコンピュータで構成してもよい。また、サーバプログラム18、クライアントプログラム14、プロキシプログラム24及びアドインAI1～AImを全て同じコンピュータにインストールすることにより、サーバ12、クライアント14及びプロキシ16を1台のコンピュータで構成してもよい。

【 0 0 6 0 】

また、上記実施の形態はCAD用の設計データを管理しているが、本発明はこれに限定されることなく、たとえば文書管理、画像管理、動画管理、スケジュール管理、地理情報

10

20

30

40

50

管理、顧客管理、販売管理、会計情報管理、生産管理、製品管理、部品管理、資産管理、施設管理、プロジェクト管理など、各種データを管理するシステムにも適用可能である。

【0061】

以上、本発明の実施の形態を説明したが、上述した実施の形態は本発明を実施するための例示に過ぎない。よって、本発明は上述した実施の形態に限定されることなく、その趣旨を逸脱しない範囲内で上述した実施の形態を適宜変形して実施することが可能である。

【図面の簡単な説明】

【0062】

【図1】本発明の実施の形態によるクライアントサーバ型CADデータ管理システムの全体構成を示す機能ブロック図である。

10

【図2】図1に示したCADデータ管理システムの初期化処理を示すフロー図である。

【図3】図3に示した初期化処理で作成されるAPI・アドインマップを示す図である。

【図4】図1に示したCADデータ管理システムにおいて、クライアントからサーバへの要求時におけるプロキシの介入処理を示すフロー図である。

【図5】図1に示したCADデータ管理システムにおいて、サーバからクライアントへの応答時におけるプロキシの介入処理を示すフロー図である。

【図6】図4に示したプロキシの介入処理のうちセキュリティアドインの例を示すフロー図である。

【図7】図5に示したプロキシの介入処理のうちログアドインの例を示すフロー図である。

20

【符号の説明】

【0063】

- 10 データ管理システム
- 12 サーバ
- 14 クライアント
- 16 プロキシ
- 18 サーバプログラム
- 22 クライアントプログラム
- 24 プロキシプログラム
- 26 API・アドインマップ
- 28 初期設定ファイル

30

【 図 1 】

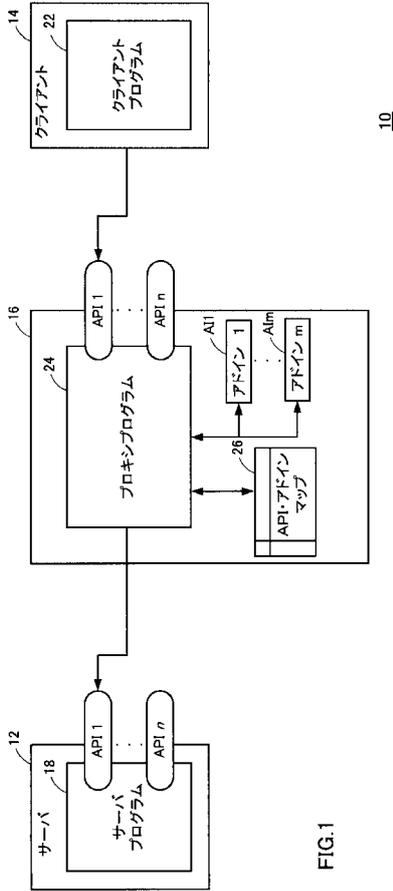
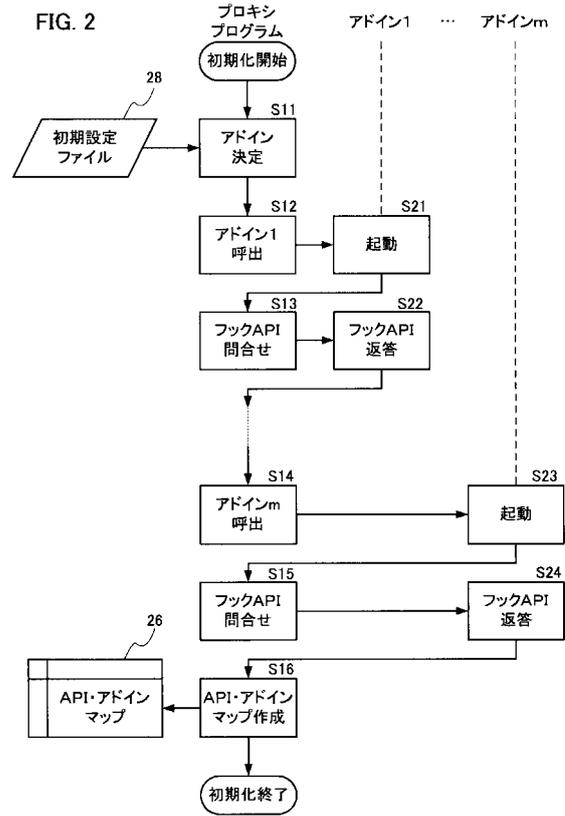


FIG.1

【 図 2 】



【 図 3 】

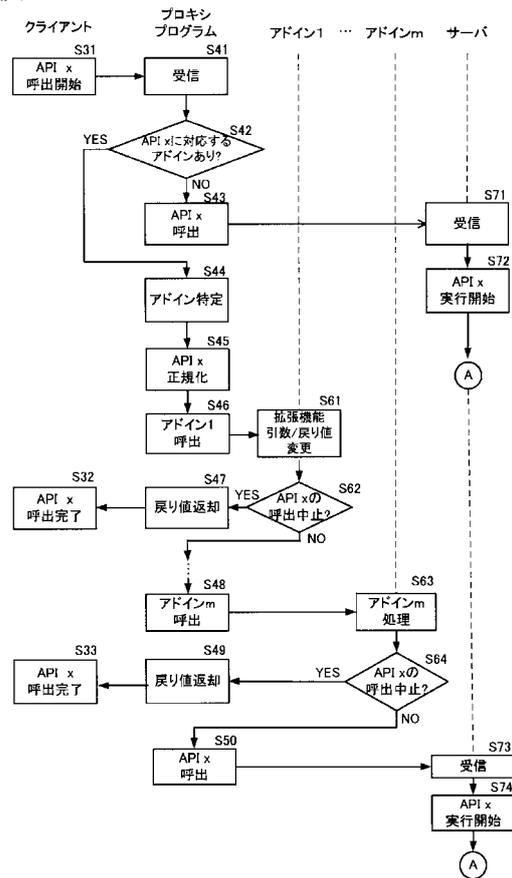
FIG.3

	API1	API2	API3	...	API _{n-1}	API _n
AI1	F	-	F		-	-
AI2	-	F	-		-	-
AI3	F	F	F		F	F
⋮						
AI _{m-1}	-	-	-		-	-
AI _m	-	-	-		F	-

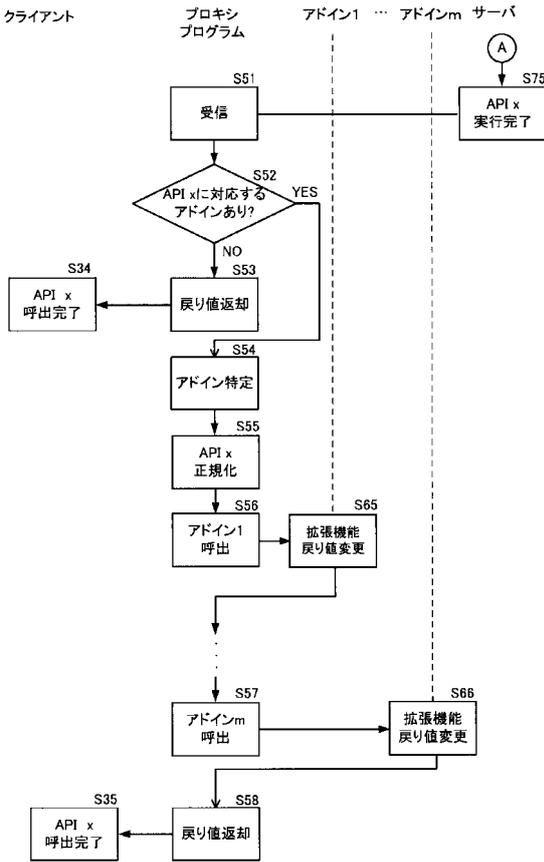
26

【 図 4 】

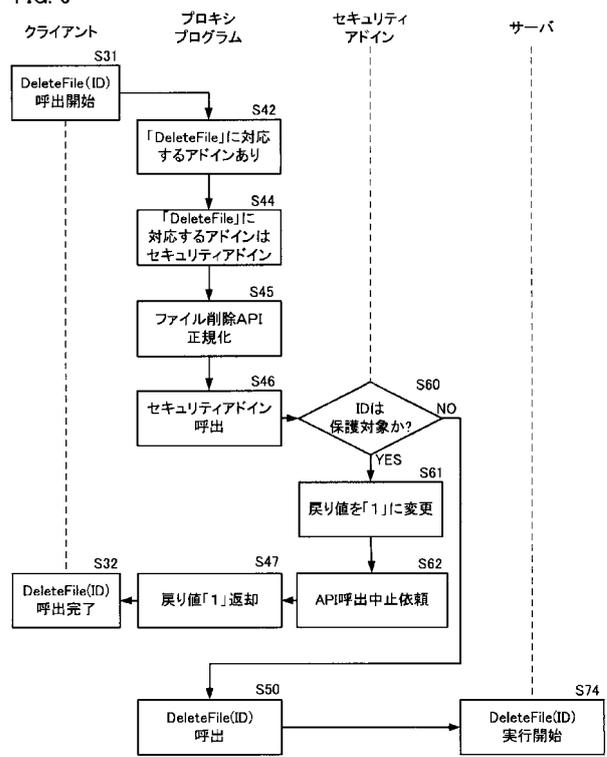
FIG. 4



【 図 5 】
FIG. 5
クライアント



【 図 6 】
FIG. 6



【 図 7 】
FIG. 7

