(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2006-72988
(P2006-72988A)
(43)公開日　平成18年3月16日(2006.3.16)

(51)Int.Cl.	F I	テーマコード（参考）
G06F 9/50 (2006.01)	G06F 9/46 465A	

審査請求 未請求　請求項の数 59　ＯＬ　外国語出願　（全 57 頁）

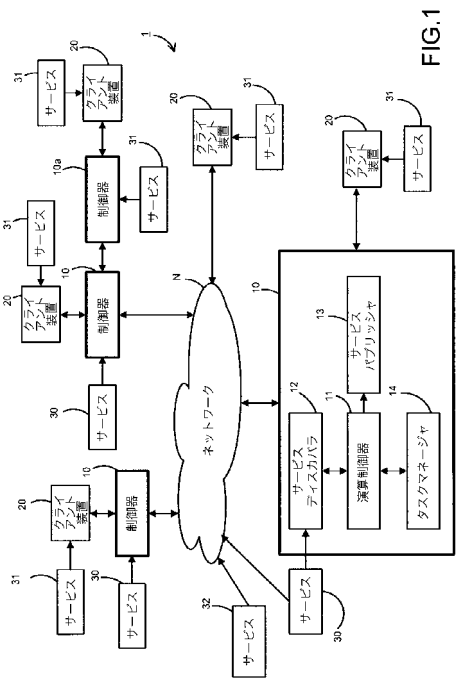
(21)出願番号　特願2005-223201 (P2005-223201)
(22)出願日　平成17年8月1日 (2005.8.1)
(31)優先権主張番号　0417101.3
(32)優先日　平成16年7月30日 (2004.7.30)
(33)優先権主張国　英国 (GB)

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二

最終頁に続く

(54)【発明の名称】タスクの実行を容易にする装置及び方法

(57)【要約】　（修正有）
【課題】ネットワークシステムにおいて、クライアント装置は、ユーザがネットワーク上で利用可能な１つ以上のサービスを発見することを目的とする。
【解決手段】制御器（１０）は、ネットワーク（N）上の位置でサービスを発見するサービスディスカバラ（１２）と、サービスディスカバラ（１２）により発見されたサービスを使用してタスクの実行を制御するタスクマネージャ（１４）とを有する。制御器（１０）は、サービスパブリッシャ（１３）を更に有し、サービスパブリッシャ（１３）は、サービスディスカバラ（１２）によりサービスを発見した結果として、そのサービスがネットワークに接続される他の制御器（１０）により発見されることを可能にする。これにより、他の制御器がその制御器を介してサービスを発見し、アクセスすることができる。
【選択図】図１



【特許請求の範囲】
【請求項１】
ユーザに対するタスクの実行を可能にするためのネットワーク接続可能な装置において、
前記ネットワーク上の位置において、サービスを発見するように動作可能なサービスディスカバラと、
前記サービスディスカバラにより発見されるサービスを使用して、タスクの実行を制御するように動作可能なタスクマネージャと、
サービスに関連する動作を制御するように動作可能な演算制御器と、
前記サービスディスカバラによりサービスを発見した結果として、サービスを前記ネットワークに接続された他の装置から発見可能にするように動作可能なサービスパブリッシャとを具備し、
前記サービスディスカバラは、異なるプロトコルを使用するサービスを発見するように構成され、且つプラグインアーキテクチャを有することを特徴とする装置。
【請求項２】
前記演算制御器は、サービスの実行要求を受信し、前記要求されたサービスが前記演算制御器により実行できる装置へのローカルサービスであるかを判定し、ローカルなサービスでない場合、前記サービスディスカバラがサービスを発見したネットワーク上の位置に、前記サービスの実行要求を渡すように動作可能であることを特徴とする請求項１に記載の装置。
【請求項３】
前記演算制御器は、サービスの実行要求を、前記タスクマネージャ及び前記ネットワークに接続される他の装置から受信するように構成されることを特徴とする請求項２に記載の装置。
【請求項４】
前記演算制御器は、前記ユーザ及び要求している前記装置のうち少なくとも一方、又はタスクマネージャに関連する少なくとも１つのセキュリティ評価に従って、サービスの実行要求が実行されるか、又は、前記サービスディスカバラが前記サービスを発見した前記ネットワーク上の位置に渡されるか否かを判定するように動作可能であることを特徴とする請求項２又は３に記載の装置。
【請求項５】
前記演算制御器は、前記タスクマネージャからのサービスの実行要求、及び他の装置からのサービス実行要求に対して、異なるセキュリティ評価を使用するように構成されることを特徴とする請求項４に記載の装置。
【請求項６】
前記演算制御器は、前記ユーザが前記装置に対してローカルであるか否か、又はサービスが別の装置により要求されているか否かに応じて、異なるセキュリティ制約を実現するように構成されることを特徴とする請求項４に記載の装置。
【請求項７】
前記サービスパブリッシャは、対応するサービスにより実現される機能を規定するサービス記述のリストを格納するように構成されることを特徴とする請求項１から６のいずれか１項に記載の装置。
【請求項８】
前記演算制御器は、前記サービスディスカバラにより発見されるサービスに対して、対応するサービスにより実現される機能を規定する複数のサービス記述を受信し格納するように構成されることを特徴とする請求項１から６のいずれか１項に記載の装置。
【請求項９】
前記サービスパブリッシャは、前記サービスディスカバラにより発見される前記サービス記述のリストを格納するように構成されることを特徴とする請求項７又は８に記載の装置。
【請求項１０】

格納された各サービス記述は、少なくとも、サービス名、前記サービスの機能の記述、並びに、サービスが必要とする入力データ、出力データ及びパラメータデータを含むことを特徴とする請求項７から９のいずれか１項に記載の装置。
【請求項１１】
前記タスクマネージャは、所定の基準に基づいて同一の機能を実行できる複数の異なる発見されたサービスから、１つのサービスを選択するように動作可能であることを特徴とする請求項１から１０のいずれか１項に記載の装置。
【請求項１２】
前記タスクマネージャは、サービスのネットワーク上の位置の指示、評価システム、必要な処理時間、サービスの現在の状態、サービス実行費用、サービス品質、及びユーザ推薦のうち少なくとも１つの基準に基づいて、同一の機能を実行できる複数の異なる発見されたサービスから、１つのサービスを選択するように動作可能であることを特徴とする請求項１から１０のいずれか１項に記載の装置。
【請求項１３】
同一のサービスが前記サービスディスカバラにより、前記ネットワーク上の異なる経路を介して２回以上発見される場合、前記演算制御器は、前記サービスへの各経路に関連する情報を格納することにより、前記演算制御器が異なる経路を介して前記サービスにアクセス可能になるように動作可能であることを特徴とする請求項１から１２のいずれか１項に記載の装置。
【請求項１４】
前記演算制御器は、同一のサービスが前記サービスディスカバラにより、前記ネットワーク上の異なる経路を介して２回以上発見され、且つ前記経路のうち１つの経路を介するサービスの実行の要求が失敗する場合、前記サービスへの別の経路を選択するように動作可能であることを特徴とする請求項１３に記載の装置。
【請求項１５】
サービスが発見されると、前記装置は、前記サービスに関連付けられ、且つネットワークを介する１つの装置から別の装置への前記サービスの位置に対するステップ数を示すホップ数を増分するように動作可能であることを特徴とする請求項１から１４のいずれか１項に記載の装置。
【請求項１６】
サービスが発見されると、前記装置は、ネットワークを介する１つの装置から別の装置への前記サービスの位置に対するステップ数を示すホップ数を増分するように動作可能であり、前記サービス記述は、ネットワークを介して採用される前記サービスの位置から前記装置への経路を示す経路情報に関連付けられ、前記サービスディスカバラがサービスを２回以上発見する場合、前記演算制御器は、関連付けられたホップ数に基づいて、経路を選択するように動作可能であることを特徴とする請求項７から１０のいずれか１項に記載の装置。
【請求項１７】
前記サービスディスカバラは、サービスディスカバリモジュール及び前記モジュールを管理するように動作可能なサービスディスカバリマネージャを含むことを特徴とする請求項１から１６のいずれか１項に記載の装置。
【請求項１８】
１つのサービスディスカバリモジュールは、ウェブサービスを発見するように構成され、別のサービスディスカバリモジュールは、ＪＩＮＩサービスを発見するように構成され、別のサービスディスカバリモジュールは、ローカルサービスを発見するように構成されることを特徴とする請求項１７に記載の装置。
【請求項１９】
前記タスクマネージャは、不完全な形式で格納され、且つ別の装置に転送されるようなタスクを提供するように構成されることを特徴とする請求項１から１８のいずれか１項に記載の装置。

【請求項２０】
前記タスクマネージャは、タスクの現在の状態及び前記タスクに関連付けられたデータを、装置間で渡されるＸＭＬコンテキスト等のコンテキストに格納するように構成されることを特徴とする請求項１９に記載の装置。
【請求項２１】
前記演算制御器は、前記装置に結合される１つ又は複数のクライアント装置が使用できないサービスを除去するように動作可能であることを特徴とする請求項１から２０のいずれか１項に記載の装置。
【請求項２２】
ユーザが前記装置とインタフェースする時に介する少なくとも１台のクライアント装置と組み合わされることを特徴とする請求項１から２１のいずれか１項に記載の装置。
【請求項２３】
少なくとも１つのサービスを格納する格納手段と組み合わされることを特徴とする請求項１から２２のいずれか１項に記載の装置。
【請求項２４】
請求項１から２０のいずれか１項に記載の複数の装置と、各装置に関連付けられた少なくとも１台のクライアント装置と、サービスを格納する複数のサービス格納手段とを具備するネットワークシステム装置において、前記装置のうち少なくともいくつかの装置は、前記ネットワークに直接接続可能であり、少なくとも１台の装置は、別の装置を介して前記ネットワークに接続可能であり、且つ前記サービス格納手段のうち少なくともいくつかの格納手段は、請求項１から２０のいずれか１項に記載の装置を介して前記ネットワークに接続されることを特徴とするネットワークシステム装置。
【請求項２５】
前記サービスは、共通アプリケーションプログラミングインタフェースを有することを特徴とする請求項１から２４のいずれか１項に記載の装置。
【請求項２６】
前記サービスは、入力サービス、出力サービス及び処理サービスを含むことを特徴とする請求項２３から２５のいずれか１項に記載の装置。
【請求項２７】
前記サービスは、入力サービス、出力サービス、処理サービス及びユーザインタフェースサービスを含むことを特徴とする請求項２３から２５のいずれか１項に記載の装置。
【請求項２８】
少なくともいくつかのサービスは、従属サービスを有することを特徴とする請求項２３から２７のいずれか１項に記載の装置。
【請求項２９】
少なくともいくつかのサービスは、パラメータに対するデフォルト値を有し、前記タスクマネージャは、サービスデフォルト値を置き換えるデフォルト値がタスクに対して提供されるように構成されることを特徴とする請求項２３から２８のいずれか１項に記載の装置。
【請求項３０】
いくつかのサービス及びいくつかのタスクのうち少なくとも１つは、少なくとも１つのパラメータに対するデフォルト値を有し、少なくとも１つのユーザインタフェースサービスは、ユーザがパラメータに対するデフォルト値を置き換える値を特定できるように構成されることを特徴とする請求項２７に記載の装置。
【請求項３１】
ユーザに対してタスクの実行を可能にする方法において、
異なるプロトコルを使用するサービスを発見するために、プラグインアーキテクチャを使用して、前記ネットワーク上の位置においてサービスを発見するステップと、
発見されたサービスを使用してタスクの実行を制御するステップと、
前記発見の結果として、発見されたサービスを前記ネットワークに接続される他の装置

から発見可能にするステップとを実行する装置を含むことを特徴とする方法。
【請求項３２】
サービスの実行要求を受信するステップと、要求された前記サービスが実行される前記装置に対してローカルであるかを判定するステップと、ローカルでない場合、前記サービスが発見された前記ネットワーク上の位置に、前記サービスの実行の要求を渡すステップとを実行する装置を更に含むことを特徴とする請求項３１に記載の方法。
【請求項３３】
サービスの実行の要求は、前記装置のタスクマネージャ及び前記ネットワークに接続される他の装置から受信されることを特徴とする請求項３１に記載の方法。
【請求項３４】
前記ユーザ及び要求している装置のうち少なくとも一方、又はタスクマネージャに関連付けられた少なくとも１つのセキュリティ評価に従って、サービスの実行要求が実行されるか又は前記サービスが発見された前記ネットワーク上の位置に渡されるか否かを判定するステップを実行する装置を更に含むことを特徴とする請求項３２又は３３に記載の方法。
【請求項３５】
異なるセキュリティ評価は、前記装置のタスクマネージャ及び他の装置からのサービスの実行要求に対して使用されることを特徴とする請求項３４に記載の方法。
【請求項３６】
異なるセキュリティ制約は、ユーザが前記装置に対してローカルか否か、又はサービスが別の装置により要求されているか否か応じて実現されることを特徴とする請求項３４に記載の方法。
【請求項３７】
対応するサービスにより実現される機能を規定するサービス記述のリストを格納する装置を更に含むことを特徴とする請求項３１から３６のいずれか１項に記載の方法。
【請求項３８】
発見されたサービスに対して、対応するサービスにより実現される機能を規定するサービス記述を受信し格納する装置を更に含むことを特徴とする請求項３１から３６のいずれか１項に記載の方法。
【請求項３９】
格納された各サービス記述は、少なくとも、サービス名、前記サービスの機能の記述、並びに、サービスが必要とする入力データ、出力データ及びパラメータデータを含むことを特徴とする請求項３７又は３８に記載の方法。
【請求項４０】
前記装置は、所定の基準に基づいて同一の機能を実行できる複数の異なる発見されたサービスから、１つのサービスを選択することを特徴とする請求項３１から３９のいずれか１項に記載の方法。
【請求項４１】
前記装置は、サービスのネットワーク上の位置の指示、評価システム、必要な処理時間、サービスの現在の状態、サービス実行費用、サービス品質、及びユーザ推薦のうち少なくとも１つの基準に基づいて同一の機能を実行できる複数の異なる発見されたサービスから、１つのサービスを選択することを特徴とする請求項３１から３９いずれか１項に記載の方法。
【請求項４２】
同一のサービスが前記ネットワーク上の異なる経路を介して２回以上発見される場合、前記装置は、前記サービスへの各経路に関連する情報を格納し、異なる経路を介して前記サービスへのアクセスを可能にすることを特徴とする請求項３１から３９のいずれか１項に記載の方法。
【請求項４３】
同一のサービスが前記ネットワーク上の異なる経路を介して２回以上発見され、且つ前記経路のうち１つの経路を介するサービスの実行要求が失敗する場合、前記装置は、前記サ

ービスへの別の経路を選択することを特徴とする請求項４２に記載の方法。
【請求項４４】
サービスが発見されると、前記装置は、ネットワークを介する１つの装置から別の装置への前記サービスの位置に対するステップ数を示すホップ数を増分することを特徴とする請求項３１から４３のいずれか１項に記載の方法。
【請求項４５】
サービスが発見されると、前記装置は、前記ネットワークを介する１つの装置から別の装置への前記サービスの位置に対するステップ数を示すホップ数を増分し、前記サービス記述は、前記ネットワークを介して採用される前記サービスの位置から前記装置への経路を示す経路情報に関連付けられ、サービスが２回以上発見される場合、前記装置は、関連付けられたホップ数に基づいて経路を選択することを特徴とする請求項３７から３９のいずれか１項に記載の方法。
【請求項４６】
前記装置は、サービスディスカバリモジュールを含み、前記方法は、前記モジュールを管理することを更に含むことを特徴とする請求項３１から４５のいずれか１項に記載の方法。
【請求項４７】
１つのサービスディスカバリモジュールは、ウェブサービスを発見し、別のサービスディスカバリモジュールは、ＪＩＮＩサービスを発見し、別のサービスディスカバリモジュールは、ローカルサービスを発見することを特徴とする請求項４６に記載の方法。
【請求項４８】
前記装置は、タスクが不完全な形式で格納され、且つ別の装置に転送されるように、タスクを構成することを特徴とする請求項３１から４７のいずれか１項に記載の方法。
【請求項４９】
前記装置は、タスクの現在の状態及び前記タスクに関連付けられたデータを、装置間で渡されるＸＭＬコンテキスト等のコンテキストに格納することを特徴とする請求項４８に記載の方法。
【請求項５０】
前記サービスは、共通アプリケーションプログラミングインタフェース（ＡＰＩ）を有することを特徴とする請求項３１から４９のいずれか１項に記載の方法。
【請求項５１】
前記サービスは、入力サービス、出力サービス及び処理サービスを含むことを特徴とする請求項３１から５０のいずれか１項に記載の方法。
【請求項５２】
前記サービスは、入力サービス、出力サービス、処理サービス及びユーザインタフェースサービスを含むことを特徴とする請求項３１から５１のいずれか１項に記載の方法。
【請求項５３】
少なくともいくつかのサービスは、従属サービスを有することを特徴とする請求項３１から５２のいずれか１項に記載の方法。
【請求項５４】
少なくともいくつかのサービスは、パラメータに対するデフォルト値を有し、前記装置により、サービスデフォルト値を置き換えるデフォルト値がタスクに対して提供可能となることを特徴とする請求項３１から５３のいずれか１項に記載の方法。
【請求項５５】
いくつかのサービス及びいくつかのタスクのうち少なくとも１つは、少なくとも１つのパラメータに対するデフォルト値を有し、ユーザは、少なくとも１つのユーザインタフェースサービスにより、パラメータに対するデフォルト値を置き換える値を特定できることを特徴とする請求項５２に記載の方法。
【請求項５６】
前記制御器に結合される１つ又は複数のクライアント装置が使用できないサービスを除去

する装置を更に含むことを特徴とする請求項３１から５５のいずれか１項に記載の方法。
【請求項５７】
請求項３１から５６のいずれか１項に記載の方法を実行するようにプロセッサをプログラミングするプログラム命令を含むことを特徴とするコンピュータプログラム製品。
【請求項５８】
請求項３１から５６のいずれか１項に記載の方法を実行するようにプロセッサをプログラミングするプログラム命令を格納することを特徴とする記憶媒体。
【請求項５９】
ユーザに対してタスクの実行を可能にするためのネットワークに接続可能な装置において、
ネットワーク上の位置においてサービスを発見するように動作可能なサービスディスカバラと、
前記サービスディスカバラにより発見されるサービスを使用して、タスクの実行を制御するように動作可能なタスクマネージャと、
サービスに関連する動作を制御するように動作可能な演算制御器と、
前記サービスディスカバラによりサービスを発見した結果として、サービスを前記ネットワークに接続される他の装置から発見可能にするように動作可能なサービスパブリッシャとを具備し、
前記サービスディスカバラは、多数のサービスディスカバリプラグインモジュールと、前記モジュールを管理するサービスディスカバリマネージャとを含み、異なるサービスディスカバリプラグインモジュールは、異なるプロトコルを使用するサービスを発見するように構成されることを特徴とする装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、タスクの実行を容易にする装置及び方法に関する。特に、本発明は、パーソナルコンピュータ、サーバ、デジタルカメラ、コピー機等の複数のプロセッサ制御装置がタスクを実行するのに必要なサービスを有する又はサービスにアクセスできる分散型システム或いはネットワークにおいて、プロセッサ制御装置によるタスクの実行を容易にする装置及び方法に関する。
【背景技術】
【０００２】
ネットワーク上の複雑なタスクを完了させるために、ユーザは、いくつかの異なるアプリケーションを使用して、異なる動作を実行する必要がある。例えば、受信者がワードプロセシングアプリケーションを使用して文書を編集できるように、ユーザが紙の文書のコピーを電子メールで送信したい場合、一般に、ユーザは、スキャナアプリケーションを使用して、紙の文書を電子形式に変換し、その後、光学的文字認識アプリケーションを使用して、電子データを電子文字データに変換する。更に、ユーザは、電子メールアプリケーションを使用して、所望する受信者に、最終電子文書を電子メールで送信する。
【０００３】
そのような複雑な動作に対するタスク単位の方法は、ユーザが実行すべき個別の動作を簡素化する。タスク単位の方法において、ネットワークの一部を構成する制御器（例えば、ユーザのプロセッサ制御装置が通信できるサーバ、又はユーザのプロセッサ制御装置）は、ユーザにより提供されたデータから、実行すべきタスクを識別し、必要なサービスから構成される所定のタスクを選択するか、又は、必要なサービスから構成される新しいタスクをアセンブルする。その後、ユーザは、制御器により入力指示されると、タスクの実行を可能にするため、ユーザ入力データを提供する必要がある。
【０００４】
上述の例において、ユーザがハードコピー文書を編集可能な形式である電子メールで送信したいことを制御器に通知する場合、制御器は、その一連のステップを実行する所定の

タスクにアクセスするか、又は、その所定の一連のステップを実行する新しいタスクをアセンブルする。いずれの場合においても、この例におけるタスクは、走査サービス、光学的文字認識サービス及び電子メールサービスを含むサービスの集合から構成される。

【０００５】

実行されるタスクの数は、制御器がアクセスできるサービスにより制限される。制御器が、ネットワークを介して遠隔からサービスにアクセスできるようにしてもよいが、他の制御器を介してネットワークにのみ結合されるサービスにアクセスすることは、不可能ではないとしても困難である。

【発明の開示】

【０００６】

１つの側面において、本発明は、タスクの実行を可能にする制御器又は装置を提供する。装置は、ネットワークに接続可能であり、ネットワークを介してサービスを発見するように動作可能なサービスディスカバラと、装置によりアクセス可能なサービスを使用して、タスクの実行を制御するように動作可能なタスクマネージャと、アクセス可能なサービスの記述又は装置により発見されるサービスの記述を発行するサービスパブリッシャとを具備する。これにより、少なくとも一部のサービスが制御器のみを介してネットワークに接続されている場合でも、ネットワークに結合される他の制御器は、制御器により発行されたサービスを認識できる。

【０００７】

１つの側面において、本発明は、タスクの実行を容易にするための制御器を提供する。制御器は、ネットワークに接続可能であり、他の制御器をサービスと同様に監視する。その結果、他の制御器は、制御器を介してサービスを発見しアクセスできる。

【０００８】

これにより、制御器は、直接認識できないサービスにアクセスでき、その結果として、更に多くのタスク又は更に柔軟性のあるタスクをユーザに提供できる。

【０００９】

一実施形態において、制御器は、別の制御器に渡されるサービスのリストを、セキュリティ制約の対象にするように構成される。例えば、制御器は、サービスにアクセスしようとしているユーザの認識、及び／又は実際のサービスの位置から要求している制御器への経路において、少なくとも１台の制御器により、サービスに対して与えられるセキュリティ制約に基づいて、別の制御器に渡されるサービスのリストを制限するように構成されてもよい。

【００１０】

制御器は、ユーザが制御器に対してローカルであるか否か、又はサービスが遠隔制御器により要求されているか否かに応じて、異なるセキュリティ制約を提供するように構成されてもよい。制御器は、サービスの記述が制御器に渡される度にホップ数を増分するように構成されてもよい。これにより、例えば、制御器が同一のサービス記述を２回以上受信する場合、制御器は、そのサービスの実際の位置に対する最も直接的な経路を判定できる。

【００１１】

制御器は、２回以上発見されたサービスの記述をキャッシュに追加するように構成されてもよい。これにより、特定のサービスに対する経路が失敗した場合、その同一のサービスは、異なる経路を介して、再び試行されてもよい。

【００１２】

一実施形態において、タスクの実行に対して利用可能なサービスは、一般に、入力サービス、出力サービス及び処理サービスを含む。利用可能なサービスは、ユーザインタフェースサービスを更に含んでもよい。これは、ユーザが制御器にアクセスする際に経由する装置のユーザインタフェースにより、ユーザが利用できるサービスが制限される可能性を回避する。あるいは、サービスとしてインタフェースをユーザに提供することにより、タスクが必要とするユーザインタフェース、又はタスク内の異なるサービスが必要とする複

数のユーザインタフェースは、必要に応じて、ユーザの装置にロードされる。これにより、専用のユーザインタフェースが、タスク内の異なるサービスに対して提供される。これは、ユーザに提供されたユーザインタフェースに情報が散乱することがないという、ユーザ装置のディスプレイサイズが相対的に小さい場合に特に重要となる利点を有する。更に、サービスとしてユーザインタフェースを提供することにより、存在するサービス及びタスクに対するユーザインタフェースが、必要に応じて、更新されることを可能にする。

【００１３】

異なるプロトコルが、異なるサービスを発見するために必要とされるであろう。一実施形態において、サービスは、共通ＡＰＩ（Application Programming Interface：アプリケーションプログラミングインタフェース）を使用し、サービスディスカバラ（service discoverer）は、プラグインアーキテクチャで構成される。これにより、サービスディスカバラは、各々が特定のプロトコルを使用するサービスを発見するように構成される１つ以上のプラグインサービスディスカバリモジュールを有する。例えば、１つのサービスディスカバリプラグインモジュールは、ウェブサービスを発見するように設計されてもよく、別のサービスディスカバリプラグインモジュールは、ＪＩＮＩサービスを発見するように設計されてもよい。更に別のサービスディスカバリプラグインモジュールは、ローカルサービス、すなわち制御器に直接接続されるサービスを発見するように設計されてもよい。プラグインアーキテクチャを使用することにより、新しいプロトコルが使用され始めると、その新しいプロトコルと共に動作するように構成されたプラグインを単に追加することにより、制御器をグレードアップ又は更新できる。新しいプロトコルが普及すると、制御器のその他の部分を変更せずに、その新しいプロトコルは、簡単にサポート及び展開される。更に、より小型の装置（デジタルカメラ等）の制御器は、制御器が必要とするプラグインのみを使用する。サービスに対して、そのようなプラグイン及び共通ＡＰＩを使用することにより、プラグインが既に展開されているプロトコルを使用するサービスは、更なる改造なしで直接使用され、新しいプロトコルを展開又は使用する場合、そのプロトコルを使用してサービスを処理するために、１つのプラグインのみを展開する必要がある。

【００１４】

タスク単位のシステムにおいて、いくつかのサービスは、ユーザにより設定されるべきあるパラメータを必要とする。しかし、ユーザは、パラメータの最適な値を常に知っているわけではない。従って、多くの場合、デフォルトパラメータが設定されていることが望ましい。

【００１５】

一実施形態において、デフォルトパラメータは、異なるレベルで提供される。サービスは、デフォルトパラメータを有し、このデフォルトパラメータは、デフォルトタスクパラメータにより置き換えられてもよい。これにより、特定のサービスが特定のタスク内で実行している時、デフォルトパラメータは、そのタスクに対して適切である。走査サービスは、例えば、平均解像度を表現するデフォルト走査解像度パラメータを有してもよく、また、写真を走査し電子メールで送信するタスクは、サービスのデフォルトパラメータを置き換える異なる解像度デフォルトパラメータを有してもよい。これにより、電子メールで送信するために写真を走査する場合、スキャナは、画像ファイルのサイズを減少させるために相対的に低い解像度で走査し、一方では、文書が光学的文字認識の対象となるタスク、例えば、ハードコピー文書を走査し、編集可能な形式で電子メールを送信する上述のタスクにおいて、走査サービスが使用される場合、タスクは、正確な光学的文字認識を可能にするため、解像度デフォルト走査パラメータを高く規定してもよい。更に、ユーザインタフェースサービスは、必要に応じて、ユーザがサービスレベルのデフォルトパラメータ又はタスクレベルのデフォルトパラメータを変化又は変更できるようにしてもよい。

【００１６】

ユーザがタスクを開始する装置は、タスク全体を完了させるのに最適な装置である必要はない。例えば、ハードコピー文書を走査し、編集可能な形式で電子メールを送信する上述のタスクの場合、ユーザは、ネットワーク化コピー機を使用して走査を実行し、その後

、情報を入力するために、パーソナルコンピュータに移動し、キーボードにアクセスしたいと考える可能性もある。更に、いくつかのサービスは、他のサービスより高い処理能力を必要とし、また、大処理容量を有する装置により、高い処理能力のサービスが完了される場合、タスクの完了は、高速化されるであろう。例えば、ネットワーク化デジタルカメラではなく、ネットワーク化パーソナルコンピュータにより特定のサービスを実行する方が、より効率的であろう。

【００１７】

一実施形態において、タスクは、不完全な形式で格納され、且つ、タスクの次の部分を完了させるために、例えば、より強力な別の制御器に転送されるように設計される。一実施形態において、タスクの現在の状態及びそれに関連付けられたデータは、コンテキストとして知られるデータ構造、例えば、ＸＭＬによるコンテキストに格納される。コンテキストは、ユーザに対して異なるクライアント装置からタスクへのアクセスを可能にする制御器間で渡される。すなわち、ユーザは、１つのクライアント装置でタスクを開始し、タスクを継続するのに更に適切な別の装置に移動できる。これは、タスク状態が後の検索のために格納されるように、又は処理を継続するために別の制御器に渡されるように、制御器が、タスクの調整及び処理を制御する場合に可能である。

【発明を実施するための最良の形態】

【００１８】

図面を参照すると、図１は、本発明を実現するネットワークシステム１の機能ブロック図を示す。

【００１９】

ネットワークシステム１は、各々が１台以上のクライアント装置２０に接続されるパーソナルコンピュータ、サーバ、他のプロセッサデバイス等の形態をとる複数の制御器１０を含む。ユーザは、制御器によりアクセス可能なサービスを使用して、制御器に対してタスクの実行を指示するための命令及び／又はデータを、クライアント装置２０により入力できる。図１に示すように、制御器１０は、ローカルサービス３０を有してもよい。更に、サービス３２は、ネットワークＮに直接接続されてもよく、サービス３１は、クライアント装置２０に直接接続されてもよい。

【００２０】

制御器１０は、ネットワークＮに直接結合されてもよく、又は１台以上の他の制御器１０を介して結合されてもよい。図１に示す例において、３台の制御器１０は、ネットワークに直接結合され、更なる制御器１０ａは、制御器のうちの別の１台を介して、ネットワークに間接的に結合される。実際のネットワーク構成は、特定の状況に依存し、例えば、３台以上又は３台未満の制御器がネットワークに直接結合されてもよいことは、当然理解されるであろう。ネットワークは、別の制御器を介してネットワークに間接的に結合される更なる制御器１０ａを含んでもよい。また、ネットワークシステムは、２台以上の他の制御器を介してネットワークに結合される更なる制御器を含んでもよい。

【００２１】

図１には示さないが、各クライアント装置２０は、制御器１０に内蔵又は実装されてもよい。更に、クライアント装置２０は、ネットワークＮを介して制御器１０と通信してもよい。

【００２２】

クライアント装置２０は、ネットワーク接続性を有するプロセッサ制御装置であり、プロセッサ制御装置は、ユーザがクライアント装置とインタフェースできるように、少なくとも１台のユーザインタフェース装置を有する。クライアント装置の例としては、パーソナルコンピュータ、コピー機、ファクシミリ、デジタルカメラ、スキャナ及び他の事務機等のネットワーク化可能な装置である。ここで、ネットワークは、オフィス環境用である。

【００２３】

ネットワークがオフィス環境以外で使用されてもよいことは、当然理解されるであろう

。従って、例えば、ネットワークがホーム環境にある場合、クライアント装置は、パーソナルコンピュータ、並びに、テレビ、ビデオレコーダ、ＤＶＤプレーヤ、及び、他のプロセッサ制御装置又は家庭内にある機器類等のネットワーク化可能な他のプロセッサ制御装置を含んでもよい。同様に、ネットワークシステムは、産業環境において使用されてもよい。この場合、１台以上のクライアント装置は、プロセッサ制御生産工場又は工作機械を含んでもよい。

【００２４】

ネットワークＮは、１つ以上の異なる種類のネットワークを含んでもよい。例えば、ネットワークＮは、ローカルエリアネットワーク１１（ＬＡＮ）、ワイドエリアネットワーク（ＷＡＮ）、イントラネット及びインターネットのうち、少なくとも１つを含んでもよい。

【００２５】

図１は、制御器１０の主な機能構成要素を示す。他の制御器１０及び１０ａの各々は、同様の機能構成要素を有する。

【００２６】

図１に示すように、制御器１０は、制御器の全体の動作を制御する演算制御器１１と、タスクの実行を制御又は管理するタスクマネージャ又はタスク解決モジュール１４を有する。更に、各制御器１０は、サービスディスカバラ１２及びサービスパブリッシャ１３を有する。サービスディスカバラ１２は、ローカルサービス３０及びネットワークＮ上のサービスの双方を発見する。サービスパブリッシャ１３は、後述するように、他の制御器１０により発見可能な制御器１０により、サービスが発見されるように構成される。すなわち、サービスパブリッシャ１３は、発見されたサービスを公示又は再発行するように構成され、これにより、それらサービスは、ネットワークＮを介して、他の制御器１０によりアクセス可能となる。

【００２７】

演算制御器１１は、制御器１０に接続される１台以上のクライアント装置２０と通信し、タスクの実行中に必要とされるサービスと通信し、且つタスクマネージャ１４により要求されたように、タスクの次のステップの動作を調整するように構成される。更に、演算制御器１１は、サービスパブリッシャ１３により発行されたサービス、すなわち、制御器１０が直接利用できないサービス要求を渡すように構成される。

【００２８】

タスクマネージャ１４は、現在利用可能なクライアント装置２０及びサービスを使用して実行されるタスクを判定し、且つ次に必要な動作、例えば、データがユーザから必要とされるか否か、又はサービスが実行されるべきか否かを判定するように構成される。

【００２９】

本実施形態において、タスクマネージャ１４は、個別のモジュールとして構成される。これにより、そのモジュールは、容易に交換可能となり、制御器１０は、クライアント装置２０に適切なタスクマネージャを有する。また、タスクマネージャ１４は、システムの変形が容易にできるように構成される。

【００３０】

上述のように、各制御器１０は、複数のローカルサービス３０にアクセスしてもよい。更に、各クライアント装置２０は、１つ以上のサービス３１に直接アクセスしてもよく、１つ以上のサービス３２は、ネットワークＮに直接結合され、全ての制御器１０によりアクセスできるようにしてもよい。

【００３１】

サービス３０、３１及び３２は、適切なメモリに格納される。サービス３０は、ネットワークシステム１のセントラルデータ記憶装置に格納されてもよく、また、サービス３０及び３１は、制御器１０又はクライアント装置２０において利用可能な任意の適切なメモリ、例えば、コンピュータの場合はハードディスクドライブ等の大容量記憶装置、又はデジタルカメラ等のクライアント装置の場合はメモリカード上に格納されてもよい。

【００３２】
本実施形態において、一般に、以下の４種類のサービスが存在する。
１）走査サービス等の入力データをタスクに提供する入力サービス
２）印刷サービス及び電子メールサービス等のタスクからデータを出力する出力サービス
３）画像操作サービス、光学的文字認識サービス等の入力データに対して、処理又は変更を行う処理サービス
４）クライアント装置にロードするためのユーザインタフェースを提供するユーザインタフェースサービス
サービスは、サブサービス（sub-services）の集合から構成されてもよい。例えば、アドレス帳サービスは、サブサービス「アドレス帳を開く」及び「アドレス帳の保存」から構成されてもよく、更にアドレス帳ユーザインタフェースサブサービスを含んでいてもよい。
【００３３】
各サービスは、以下の４つのメソッドを実装する同一の共通ＡＰＩ（Application Programming Interface：アプリケーションプログラミングインタフェース）を使用する。
１．Get description
２．Get status
３．Run service
４．Get user interface(s)
共通ＡＰＩは、制御器１０がサービスを発見し、それらサービスの記述及び状態を取得できるようにする。更に、タスクマネージャ１４が、サービスが要求されたことを示す場合、共通ＡＰＩは、制御器１０が関連するユーザインタフェースを取得し、サービスを実行できるようにする。
【００３４】
図２ａは、サービス記述の概略図１００を示す。
【００３５】
サービス記述は、以下の構成要素を含む。
１．サービス名
２．ＩＤ
３．種類
４．入力
５．出力
６．パラメータ
７．機能
また、サービス記述は、任意に構成要素「分類」を含む。
【００３６】
サービス名は、サービスに固有であるため、例えば、ネットワークＮ上に同一のサービスを実行する２台以上の制御器１０又はクライアント装置２０が存在する場合、それら制御器１０又はクライアント装置２０は、同一のサービス名が割り当てられる。しかし、ＩＤは、特定のサービス、及びそのサービスを実際に実行する制御器又はクライアント装置に固有であり、通し番号又は他の固有のコードの形態をとってもよい。
【００３７】
構成要素「種類」は、サービスの種類を規定し、サービスが走査、電子メール、ＯＣＲ、印刷、画像操作等を実行する動作又は手順を記述する。
【００３８】
任意の構成要素「分類」は、より下位レベルのサービスの記述を示し、サービスを介して利用可能な機能を識別する。例えば、サービスの種類が「画像操作」の場合、分類は、サービスにより提供される画像操作の特性を特定する。画像操作サービスに対する分類には、例えば、サイズ変更、赤目除去、回転、パノラマ作成等がある。
【００３９】

構成要素「入力」は、サービスに渡されるデータの種類、及びそのデータに対して必要とされる形式を規定する。また、構成要素「出力」は、サービスが提供するデータの種類、及びその出力データが有する形式を規定する。

【００４０】

構成要素「パラメータ」は、サービスに渡され、且つサービスがデータに対して行うことに影響を与える1つ以上の値を含む。サービス記述は、１つ以上のパラメータに対して、１つのデフォルト値を含んでもよい。印刷サービスに対するパラメータの例として、印刷数に対する値がある。

【００４１】

構成要素「機能」は、サービスが提供できる内容についての記述を提供する。例えば、構成要素「機能」は、走査サービス又は印刷サービスのｄｐｉ単位の最大解像度を規定してもよい。

【００４２】

入力、出力、パラメータの各々は、データの特性（例えば、画像又は文書）を記述するＩＤ、データが属する種別を識別する種類、及びデータが採用する正確な形式を識別する形式を有する。例えば、種類が「ＭＩＭＥ」である場合、形式は、画像／ｐｎｇ又はアプリケーション／ｗｏｒｄであり、種類が「ｓｉｍｐｌｅ」である場合、形式は、「ｓｔｒｉｎｇ」又は「ｉｎｔ」となるであろう。

【００４３】

更に、入力、出力及びパラメータは、これらのプロパティに対して、制約又は制限を規定してもよい。これは、入力データの最大サイズ、最大解像度又は最小解像度が規定されてもよいからである。

【００４４】

サービス記述は、サービスが適切であり、且つ特定のタスクに対して必要な動作を実行できるかを判定するのに必要な情報を、制御器１０に提供する。

【００４５】

図２は、制御器１０の一例の更に詳細な機能ブロック図を示す。

【００４６】

図２に示すように、サービスディスカバラ１２は、複数のプラグイン１６（示される例では、３つ）を受信するように構成されるサービスディスカバリマネージャ１５を含む。各プラグイン１６は、あるプロトコルに従って動作するサービスを発見できるように構成される。図に示すように、プラグイン１６は、例えば、ＵＤＤＩ（Universal Description Discovery and Integration）プロトコル及びＳＯＡＰ（Simple Object Access Protocol）通信プロトコルに従って動作するウェブサービスを発見するように構成されるプラグイン、ＪＩＮＩサービスを発見するように構成されるプラグイン、及びローカルサービスを発見するように構成されるプラグインを含んでもよい。図２は、ローカルサービス３０を発見したローカルプラグイン１６を示す。共通ＡＰＩを示すブロック１９を使用して、サービスが共通ＡＰＩを使用又は実装することを、図２に示す。

【００４７】

プラグインアーキテクチャを使用することにより、新しいディスカバリシステム及びプロトコルが開発されると、新しいプラグイン１６は、これらのプロトコルを使用してサービスを発見できるように設計される。更に、制御器１０は、制御器１０と通信するクライアント２０により必要とされるプラグインのみを使用する必要がある。例えば、制御器１０がデジタルカメラ等の専用のプロセッサ制御装置とのみ通信している場合、デジタルカメラが使用できるサービスに対応するプラグインのみが提供される。

【００４８】

プラグインアーキテクチャのみでなく、ネットワークシステムにより使用可能なサービスは、全て、同一の共通ＡＰＩを使用するように構成される。新しいサービスにより必要とされるプロトコルに対して、プラグインが存在すれば、共通ＡＰＩを使用することにより、そのサービスは直接使用可能となる。

【００４９】
図２に示すように、演算制御器１１は、サービスキャッシュ１７を保持し、発見されたサービスのサービス記述を保持する。更に、サービスパブリッシャ１３は、制御器１０により発見された全てのサービスのサービス記述を含むサービスリスト１８を格納する。また、サービスパブリッシャ１３は、共通ＡＰＩ（ブロック１９により図示される）を使用して、ネットワークＮ及び直接接続されるクライアント装置２０と通信するように構成される。これにより、ネットワークＮ上の他の制御器１０及び接続されるクライアント装置２０は、制御器をサービスとして認識できる。
【００５０】
制御器１０は、パーソナルコンピュータ、サーバ、又は他のプロセッサデバイス等の演算装置をプログラムすることにより実現されてもよい。図３は、複数のプログラム命令によりプログラムされ、制御器１０に提供される演算装置５０の一例の機能ブロック図を示す。
【００５１】
図３に示すように、演算装置５０は、ＲＯＭ及び／又はＲＡＭ、並びにハードディスクデバイス等の大容量記憶装置５２の形態をとる連想メモリ５１を有するプロセッサ５００を含む。演算装置は、例えばＣＤ-ＲＯＭ、ＤＶＤ、フロッピディスク等の取外し可能な媒体５４を受け入れるための取外し可能な媒体装置５３を更に含む。更に、演算装置は、複数のユーザインタフェース装置５６を含む。図に示すように、ユーザインタフェース装置５６は、マウス、キーボード５６ｂ、マイクロフォン５６ｃ、スピーカ５６ｄ、ディスプレイ５６ｅ及びプリンタ５６ｆ等のポインティングデバイス５６ａを含む。演算装置は、ネットワークＮを介して通信できるようにするためのネットワークカード及び／又はＭＯＤＥＭ等の１つ以上の通信装置５７と、制御器１０に直接リンクするクライアント装置２０及びサービス３０等の外部装置と通信できるようにするための１つ以上の入出力インタフェース５８とを更に含む。
【００５２】
演算装置は、以下のうちの少なくとも１つによりプログラムされる。
【００５３】
メモリ５１又は大容量記憶装置５２に予め格納されたプログラム命令、
通信装置５７又はＩ／Ｏインタフェース５８を介して信号として提供されるプログラム命令、
取外し可能な媒体装置５３を介して受け入れた取外し可能な媒体５４、又はＵＳＢポート等のＩ／Ｏインタフェースを介して演算装置に接続可能な携帯用データ記憶装置からダウンロードされるプログラム命令、及び
キーボード５６ｂを使用してユーザにより直接入力されるプログラム命令。
【００５４】
図３は、メモリ５１がサービスディスカバリマネージャ１５を実現するディスカバリマネージャモジュール５１ａ、プラグイン１６を実現するプラグインモジュール５１ｂ、演算制御器１１を実現する演算制御器モジュール５１ｃ、タスクマネージャ１４を実現するタスクマネージャモジュール５１ｄ、及びサービスパブリッシャ１３を実現するサービスパブリッシャモジュール５１ｅを含むようにプログラムされた演算装置を示す。
【００５５】
プロセッサ制御装置であるクライアント装置２０は、連想メモリを有するプロセッサ、Ｉ／Ｏインタフェース、及びある形態のユーザインタフェース装置を更に有することが、理解されるであろう。ここで、ユーザインタフェース装置は、パーソナルコンピュータの場合、図３に示される装置と類似し、デジタルカメラの場合、更に制限され、一般に、小型のディスプレイ及び表示されたメニューからの項目選択を可能にするユーザ制御装置から構成される。
【００５６】
次に、制御器１０が複数のサービスに基づくタスクの実行を可能にする方法について、

説明する。
【００５７】
タスクは、特定のタスクに依存して、１つ以上のサービスから構成される。共通タスク、使用されているタスク、又は頻繁に使用されるタスクは、タスクマネージャ１４により格納されてもよい。他のタスクは、ユーザから受信した入力情報に従って、タスクマネージャ１４により、利用可能なサービスからアセンブルされてもよい。
【００５８】
タスクが実行されるために、当然、そのタスクは、最初にユーザにより開始される必要がある。これを容易にするために、クライアント装置２０は、ユーザに「何を行いたいですか？」等の質問を表示する基本タスクユーザインタフェースが提供されてもよく、また、現在利用可能なタスクを規定する複数のオプションを、ユーザに提供してもよい。一般に、利用可能なタスクの特性は、特定のクライアント装置に依存する。例として、クライアント装置がネットワーク化コピー機である場合、タスクのオプションは、以下を含んでもよい。
走査
走査及び電子メール
走査、ＯＣＲ及び電子メール
別の例として、基本タスクユーザインタフェースは、表示されたメニューからオプションを選択することにより、ユーザがタスクを規定できるようにしてもよい。例えば、ユーザがハードコピー文書を走査し、それを編集可能な形式で電子メールで送信したい場合、ユーザは、表示されているオプション「走査」、「光学的文字認識」及び「電子メール」を、メニューから選択してもよい。
【００５９】
図４は、タスクマネージャにより実行されるステップを示すフローチャートである。
【００６０】
上述のように、制御器１０は、発見されたサービスに対する記述を格納する。図４のＳ１において、タスクマネージャ１４は、利用可能な発見されたサービス、クライアント装置、及びユーザから、そのクライアント装置及びユーザが利用可能なタスクを判定する。
【００６１】
Ｓ２において、タスクマネージャは、ユーザに対して、そのユーザが利用可能なタスク／オプションのリストをクライアント装置２０に提示させる。利用可能なタスクの一例は、「ハードコピー文書を走査し、光学的文字認識を実行し、且つ編集可能な形式でその文書を電子メールで送信する」であってもよい。
【００６２】
Ｓ３において、ユーザがクライアント装置で提供されるリストからタスクを選択すると、タスクマネージャは、Ｓ４において、格納されたサービス記述の構成要素「種類」をチェックすることにより、タスクが必要とする最初のサービスを識別し、タスクの要求に合致する構成要素「種類」を判定し、且つその中から、タスクが受け入れ可能な入力パラメータ、出力パラメータ、機能パラメータ及び分類パラメータを有するサービスを判定する。
【００６３】
２つ以上のサービスが、要求に合致してもよい。２つ以上のサービスが合致する場合、Ｓ４において、タスクマネージャは、以下に説明する所定の基準に従って、それら合致するサービスのうちの１つを選択する。
【００６４】
タスクの最初の手順又はステップに対して、１つのサービスが選択されると、Ｓ５において、タスクマネージャ１４は、ユーザ入力が必要とされるか否かを判定する。必要とされる場合、タスクマネージャ１４は、Ｓ６において、必要なユーザ入力を取得するために、ユーザに対して適切なユーザインタフェースを、クライアント装置により提供させる。
【００６５】

Ｓ５においてユーザ入力が必要とされない場合、又はユーザ入力がＳ６において既に取得されている場合、タスクマネージャは、Ｓ７において、必要なサービスを実行するように、演算制御器１１に要求する。

【００６６】

Ｓ８において、タスクマネージャは、タスクが完了したかをチェックする。すなわち、タスクが別のサービスを必要とする更なる手順を含むかをチェックする。

【００６７】

タスクマネージャが、タスクは別のサービスを必要とする更なる手順を含むと判定する場合、ステップＳ４～Ｓ７が繰り返される。ステップ８において、タスクマネージャは、タスクを完了するためにサービスの実行を必要とする更なる手順が存在するかを、再びチェックする。Ｓ８において、タスクマネージャがサービスの実行を要求する更なる手順が存在しないと判定すると、手順は終了する。

【００６８】

図５は、サービスが発見された時に実行されるステップを示す。

【００６９】

演算制御器１１は、サービスディスカバラ１２を開始すると、Discovery Listenerをサービスディスカバリマネージャ１５に登録する。サービスディスカバラ１２は、プラグイン１６の位置を特定すると、プラグイン１６を開始し、サービスディスカバリマネージャ１５をプラグイン１６と共に登録する。サービス３０、３１、３２がプラグイン１６により発見されると、プラグイン１６は、サービスディスカバリマネージャ１５に通知し、サービスディスカバリマネージャ１５は、演算制御器１１に通知する。

【００７０】

新しいサービスが発見されると、図５のＳ１１において、演算制御器１１は、サービス記述を受信し、サービス記述に関連する経路情報を更新する（例えば、経路アドレスに制御器ネットワークアドレスを追加することにより、又はホップ数に１加えることにより更新する。ここで、「ホップ」は、ネットワーク上の１つの制御器から次の制御器に対するステップ又は経路である）。更に、演算制御器１１は、関連する経路情報と共にサービス記述をサービスキャッシュ１７に格納し、サービスリスト１８へ格納するために、そのサービス記述を関連する経路情報と共に、サービスパブリッシャ１３に渡す。別の方法として、ホップ数は、サービスパブリッシャ１３により更新されてもよい。

【００７１】

各制御器１０のサービスパブリッシャ１３が、共通ＡＰＩ１９を使用又は実装することにより、発見されたサービスを再発行する（すなわち、ネットワークＮ上の他の制御器へのアクセスを可能にする）ため、同一のサービスは、複数の異なる経路を介して、制御器１０により発見されてもよい。サービスが２回以上発見される場合、サービス記述は、ステップＳ１２において、サービスキャッシュ１７に追加される。これにより、そのサービスを実行する後続の要求が失敗する場合、演算制御器１１は、他の経路を介して、又は他の複数の経路のうちの１つを介して、サービスへのアクセスを試みる。

【００７２】

図６は、Ｓ２０においてタスクマネージャ１４がサービスの実行要求を発行する時に、演算制御器１１により実行される動作を示すフローチャートである。Ｓ２１において、演算制御器１１は、Ｓ２１において選択されたサービスのサービス記述のrun serviceメソッドを実装し、サービスを実行させる。サービスがローカルサービスでない場合、制御器１０は、サービスの実行要求を、そのサービスが発見された制御器に渡す。その制御器がその特定のサービスを実際に実行する制御器でない場合、サービスの実行要求がその特定のサービスを実際に実行する制御器に到達するまで、制御器は、順次、サービスが発見された制御器に、そのサービスの実行要求を渡していく。その特定のサービスを実際に実行する制御器は、サービスを呼び出し、要求している制御器１０に結果を戻す。

【００７３】

Ｓ２２において、例えば、経路パスに沿った１つ以上のネットワークアドレスが動作不

能であるため、又はサービスが現在使用中であるため等の理由により、サービスの実行要求が失敗する場合、演算制御器１１は、Ｓ２３において、同一のサービスに対する別の経路があるかを判定し、別の経路がある場合、Ｓ２４において他の経路を選択し、ステップＳ２１～Ｓ２３を繰り返す。

【００７４】

Ｓ２３において、ＮＯの場合、すなわち、サービスが失敗し、まだ探索されていない他の経路がない場合、演算制御器は、Ｓ２５において、別の受け入れ可能なサービスがあるかをチェックし、サービスがある場合、ステップＳ２６において他のサービスを選択し、ステップＳ２１～Ｓ２６を繰り返す。

【００７５】

特定の手順を実行するために、２つ以上のサービスが利用可能である場合、図４のＳ４において説明したように、そのサービスのうち１つのサービスが、所定の基準に基づいて選択されてもよい。これらの基準は以下のうち少なくとも１つを含む。

１．サービスを実行する制御器に到達するのに必要なホップ数。これは、サービスの信頼性及びサービスにアクセスするのにかかる時間を判定できるからである

２．評価システム

３．必要な処理時間

４．サービスの現在の状態

５．費用

６．品質

７．システム上のユーザからの推薦

適切なユーザインタフェースを介してユーザが入力する情報により、上述のうち２つ以上を選択してもよい。

【００７６】

上述に加え、特定のサービスのみを通すように、制御器１０は、制御器１０がサービスを供給するクライアント装置２０により、特殊化されてもよい。これは、特定のプロトコルに準拠するサービスのみが発見されるように、プラグインを選択することにより達成されてもよく、又はサービス記述がある要求に合致するサービスのみを通す演算制御器１１により達成されてもよい。例えば、クライアント装置がデジタルカメラである場合、制御器１０は、デジタルカメラに適用可能なサービスのみ、すなわち、例えば文書処理サービスを除くサービスを通すように構成される。

【００７７】

演算制御器１１は、要求している制御器にサービスが渡されるか否かを判定するセキュリティ設定を更に有してもよい。図７は、これが実現される１つの方法を示すフローチャートである。Ｓ３０において、制御器１０は、サービスディスカバリ要求を受信し、その要求は、データに添付される。このデータは、要求している制御器を識別し、更に、要求しているユーザを識別してもよい。

【００７８】

Ｓ３１において、演算制御器１１は、要求している制御器又はタスクマネージャ、及び／又はユーザに対するセキュリティ評価が要求されたサービスに対するアクセスを可能とするかを判定する。可能でない場合、演算制御器１１は、要求しているサービスディスカバラ１２に、サービスが利用不可能であることを通知する。

【００７９】

演算制御器１１は、ローカルユーザ及び遠隔ユーザに対して、異なるセキュリティ評価を有してもよい。

【００８０】

演算制御器１１は、要求している制御器及び／又はユーザが適切なセキュリティ評価を有していると判定する場合、Ｓ３３において、サービスが再発行されたサービスであるかをチェックする。再発行されたサービスではない場合、Ｓ３４において、サービスを実行させる。しかし、サービスが再発行されたサービスである場合、Ｓ３５において、演算制

御器は、サービスが発見された制御器に、そのサービスに対する要求を渡す。その制御器は、図７に示す手順を繰り返し、その要求は、要求されたサービスを実行する制御器に到達するまで、制御器から制御器に渡される。

【００８１】

図７に示すように、制御器は、サービスが再発行されたサービスであるか否かを判定する前に、セキュリティ評価をチェックするように構成される。これにより、各制御器は、遠隔サービスに対するセキュリティ評価を有することができる。別の方法として、ステップＳ３３及びＳ３１を逆にして、演算制御器１１が、サービスが再発行されたサービスであるかを最初にチェックし、再発行されたサービスである場合、セキュリティ評価をチェックせずに、サービスが発見された制御器にその要求を渡すようにしてもよい。これにより、特定の制御器及び／又はユーザがサービスへのアクセス権を付与されるか否かが、最初にそのサービスを発見した制御器単独で判定される。

【００８２】

図４は、タスクの１つの実行方法を示す。図８は、ユーザがタスクを選択した後か、又は、例えば、ハードコピー文書を走査し、光学的文字認識を実行し、その文書を編集可能な形式で電子メールで送信するというタスクの要求をユーザが指示した後に、タスクが実行される別の方法を示す。

【００８３】

上述のように、制御器１０は、発見されたサービスに対する記述を格納する。

【００８４】

図８のＳ４０において、タスクマネージャ１４は、サービスキャッシュ１７に格納されている記述から、タスクの最初の手順又はステップに対する要求に合致するサービスを識別するように、演算制御器１１に要求する。

【００８５】

タスクマネージャ１４は、格納されているサービス記述の構成要素「種類」をチェックし、構成要素「種類」がタスクマネージャの要求に合致するかを判定する。合致する場合、タスクマネージャ１４は、そのサービス記述の入力パラメータ、出力パラメータ、機能パラメータ及び分類パラメータが対象タスクにとって受け入れ可能であるかを判定する。

【００８６】

Ｓ４１において、演算制御器１１が、適切なサービスは識別されなかったと報告する場合、Ｓ４２において、タスクマネージャ１４は、演算制御器１１に対して、サービスディスカバラ１２にサービスの位置を特定するように指示することを要求する。

【００８７】

２つ以上のサービスが、タスク要求に合致してもよい。従って、Ｓ４３において、タスクマネージャ１４は、２つ以上のサービスが識別されたかを判定する。２つ以上のサービスが識別された場合、以下に説明する所定の基準に基づいて、１つのサービスを選択する。

【００８８】

タスクの最初の手順又はステップに対して、１つのサービスが選択されると、Ｓ４４において、タスクマネージャ１４は、タスクが別のサービスを必要とする更なる手順を含むかを判定する。Ｓ４４においてＹＥＳの場合、ステップＳ４０～Ｓ４３を繰り返す。

【００８９】

タスクを完了するために必要な全ての手順に対して、複数のサービスが発見されると、Ｓ４５において、タスクマネージャ１４は、次の動作又はタスクにより必要とされる手順を判定し、且つそれに応じて動作するように演算制御器１１に要求する。Ｓ４５において、タスクマネージャは、ユーザ入力が要求され、且つユーザ入力が新しいユーザインタフェースを含まないと判定すると、演算制御器に対して、クライアント装置２０と通信して、必要なデータの入力をユーザに指示するように要求する。あるいは、タスクマネージャは、必要とされる次の動作は実行すべきサービスに対するものであると判定すると、演算制御器１１に対して、必要なサービスを実行するように要求する。その必要なサービスは

、場合によっては、ユーザインタフェースサービス等のサブサービスを含み、この場合、演算制御器１１は、最初に、必要なユーザインタフェースをクライアント装置２０にダウンロードする。

【００９０】

タスクマネージャがＳ４６においてタスクが完了したと判定するまで、すなわち、タスクを完了するために必要な全てのサービスが実行されるまで、ステップＳ４５は、繰り返される。

【００９１】

上述のように、各制御器は、共通ＡＰＩを使用又は実装して、発見されたサービスを再発行する。これにより、他の制御器に関する限り、各制御器は、サービスとして見られる。利用可能なサービスは、入力サービス、出力サービス、処理サービス及びユーザインターフェースサービスを含んでもよい。サービスとしてユーザインタフェースを提供することにより、ユーザインタフェースは、必要に応じて、クライアント装置にロードされ実行される。従って、専用のユーザ入力サービスは、特定のサービスに提供され、更に、特定のクライアント装置に対するユーザインタフェースは、容易に更新及び／又は変更される。

【００９２】

上述のように、サービスは、そのサービスにより提供される動作に対するデフォルトパラメータを含んでもよい。例えば、サービスが走査サービスである場合、デフォルト走査解像度が規定されてもよい。同様に、サービスが印刷サービスである場合、白黒、片面等のデフォルト印刷特性が規定されてもよい。いくつかのタスクにおいて、サービスに対して、特定のパラメータが必要とされるであろう。例えば、ハードコピー文書を走査し、編集可能な形式で電子メールを送信するタスクの場合、光学的文字認識ソフトウェアは、走査サービスの通常のデフォルト走査解像度よりも高い走査解像度を必要としてもよい。そのような場合に対処するため、タスクマネージャ１４は、タスクに１つ以上のデフォルトパラメータを提供するように構成されてもよく、そのタスクは、タスクのデフォルトパラメータがサービスのパラメータを置き換えるように構成されてもよい。上述のタスクを例にすると、これにより、タスクマネージャ１４は、タスクに対して、走査サービスのデフォルト走査解像度パラメータよりも高いデフォルト走査解像度パラメータを規定できる。一方、写真を走査し電子メールで送信するタスクの場合、タスクマネージャ１４は、走査サービスのデフォルトパラメータを置き換える低い解像度走査パラメータを規定してもよい。これにより、大きすぎず、電子メールで送信するのに適切な画像ファイルが、走査サービスにより生成される。更に、タスクに対するユーザインタフェースサービス又はタスク内のサービスにより、ユーザが、特定のユーザ要求に従って、サービスレベルのデフォルトパラメータ又はタスクレベルのデフォルトパラメータを変化又は変更できるようにしてもよい。

【００９３】

これまでに説明したように、タスクが規定され、且つ必要なサービスがアセンブルされると、タスクが開始された制御器１０の演算制御器１１及びタスクマネージャ１４は、最初にそれらサービスを発見した制御器が実行しているタスクが必要とするサービスと連携させて、タスクを実行する。

【００９４】

ユーザのローカル制御器は、ネットワークによりサービスが提供される物理的な位置の移動に応じて変更されてもよいが、これは、ユーザにとって常に便利であるとは限らない。タスク内のあるサービスを実行する際には、あるクライアント装置が、より便利であることもある。例えば、ハードコピー文書を走査し、編集可能な形式で電子メールを送信する上述のタスクの場合、ネットワーク化コピー機を使用する方が、パーソナルコンピュータに付属のスキャナを使用する方よりも高速であるため、ユーザは、走査を実行するのにネットワーク化コピー機を使用したいが、電子メールのメッセージを入力するために、パーソナルコンピュータに移動し、キーボードにアクセスしたいと考えるだろう。更に、い

くつかのサービスは、他のサービスよりも高い処理能力を必要とするであろうし、多くの処理容量を有する装置により高い処理能力のサービスが完了される場合、タスクの完了は、高速化されるであろう。例えば、特定のサービスが、ネットワーク化デジタルカメラよりも、ネットワーク化パーソナルコンピュータにより実行される方がより効率的であろう。ここで、これら双方の装置は、同一又は類似のサービスを実行する。

【００９５】

一実施形態において、タスクマネージャ１４は、タスクを提供するように構成され、これにより、タスクは、不完全な形式で格納され、且つ、タスクの次の部分を完了させるために、例えば、より強力な別の制御器に転送される。

【００９６】

一実施形態において、タスクマネージャ１４は、タスクの現在の状態及び関連するデータを、制御器間で渡されるコンテキストに格納する。尚、この制御器により、ユーザは、異なるクライアント装置からタスクにアクセスできる。すなわち、ユーザは、１つのクライアント装置でタスクを開始し、タスクを継続するのに更に適切な別の制御器に移動できる。これは、タスクデータが後の検索のために格納されるように、又は処理を継続するために別の制御器に渡されるように、ユーザの現在のクライアント装置にローカルな制御器がタスクの調整及び処理を制御する場合に可能となる。

【００９７】

一実施形態において、サービス及びタスクは、ＸＭＬ（Extensible Mark-up Language：拡張マークアップ言語）を使用して規定され、タスクは、そのタスクの一部が制御器間で渡されるタスクのコンテキスト有効部分の中に格納される。

【００９８】

付録１は、一連の画像からウェブギャラリを作成するウェブギャラリサービスに対するサービス記述の一例を示す。省略符号は、省略されたデータを示す。付録１から分かるように、サービス名は、「web gallery」であり、サービスの種類は、「synthesize」である。このサービス記述は、必要な入力、複数の出力及び複数のパラメータを規定する。付録１から分かるように、入力は、ＩＤ「image」を有し、ＪＰＥＧ形式で、その種類がＭＩＭＥである必要がある。一方で、ＩＤ「HTML」の出力は、ＨＴＭＬ形式で、その種類がＭＩＭＥである必要があり、ＩＤ「fullimage」及びＩＤ「thumbimage」の出力は、その種類がＭＩＭＥで、ＪＰＥＧ形式である必要がある。この例におけるパラメータは、タイトル、ウェブファイル名、ウェブ画像ディレクトリ及びウェブサムネール画像プレフィックスが、「string」形式であり、残りのパラメータが、「int」形式又は整数である各々の種類が全て「simple」であるタイトル、１行毎のサムネール画像数、サムネール画像の幅、サムネール画像の高さ、ウェブファイル名、ウェブ画像ディレクトリ及びウェブサムネール画像プレフィックスである。

【００９９】

付録２は、タスク記述の一例を示す。この例において、タスクの目的は、文書を走査し、電子メールを介して、それを送信することであり、そのタスク名は、「Send OCR」である。付録２から分かるように、タスクは、サービスＩＤが「address」のアドレス帳サービス、サービスＩＤが「service 2」のスキャナサービス、サービスＩＤが「convert」のＰＤＦ変換サービス、サービスＩＤが「service 0」の電子メールサービス、及びサービスＩＤが「service 1」のＯＣＲサービスから構成される。タスクは、ＩＤ「service 0」、「service 1」及び「convert」のサービスが実行されている時に、クライアント装置にダウンロードされる「OCR email UI」という名前のユーザインタフェースサービスを更に含む。

【０１００】

このタスクがタスクマネージャにより実現されると、タスクマネージャ１４は、まず、ユーザが電子メールの指定受信者名及びそのアドレスを選択できるように、アドレス帳ユーザインタフェースサービスが実行され、クライアント装置にダウンロードされるように、演算制御器１１に対して要求する。指定受信者名及びそのアドレスがユーザにより選択

又は入力されると、タスクマネージャ１４は、スキャナサービスが実行されるように、演算制御器１１に対して要求する。ユーザによりスキャナに置かれたハードコピー文書は、デフォルト解像度で、この例では３００ＤＰＩで、スキャナサービスにより走査され、「convert」サービスへの入力のために、ＩＤ「image」の出力データを提供する。走査に必要なパラメータ、例えば、走査解像度がタスクのデフォルト値により与えられるため、スキャナサービスは、ユーザインタフェースの実行を必要としない。タスクマネージャ１４は、スキャナサービスが出力データを提供したと判定すると、残りのサービスに対して、ユーザからの必要なデータの入力を可能にするために、ユーザインタフェースサービス「OCR Email UI」が実行され、且つクライアント装置にダウンロードされるように、演算制御器１１に要求する。タスクマネージャ１４は、ユーザが必要な情報を入力したと判定すると、「convert」サービス、又は「OCR」サービス、又はその双方のサービスを実行するように、演算制御器１１に要求する。ここで、ユーザが入力した必要な情報の判定は、ユーザがＯＣＲされた文書を電子メール受信者に送信したいのか、且つ／又は画像から変換されたＰＤＦファイルで送信したいのかという判定である。タスクマネージャ１４は、「convert」サービス及び「OCR」サービスが動作を完了したと判定すると、必要に応じて、文書が指定受信者に電子メールで送信されるように、電子メールサービスを実行することを、演算制御器１１に要求する。

【０１０１】

付録２から理解されるように、演算制御器１１及びタスクマネージャ１４は、タスクの実行を調整する。これにより、各サービスは、必要に応じて呼び出され、ある目標サービスＩＤ及び目標入力ＩＤを有する目標サービスに対して、出力データを提供する。

【０１０２】

サービスディスカバラ１２は、演算制御器により指示される場合にのみ、サービスを発見してもよい。別の方法として、サービスディスカバラ１２は、システム上の新しいサービスを継続的にチェックし、新しいサービスが発見された時に、演算制御器１１に警告を与えるように構成されてもよい。これにより、ネットワーク上で発見可能な全てのサービスが、タスクマネージャにとって即時に利用可能となるという利点を有するであろう。しかし、ネットワークが多くのサービスを提供する場合、これら全てのサービスの記述を保持するために相当なメモリ容量を必要とし、制御器の資源が制限される場合には、望ましくない。当然、ネットワーク上の１台以上の制御器は、要求された時にのみ、サービスを発見するように構成されてもよく、ネットワーク上の制御器のうち１台以上の他の制御器は、制御器の性能及び要求に応じて、継続的にサービスを発見するように構成されてもよい。

【０１０３】

上述の例において、ユーザは、単一のクライアント装置を介してシステムにインタフェースする。上述のように、ある状況において、タスクは、ネットワークシステム内で移植されることが望ましい。これは、コンテキストとして知られるデータ構造にタスクを格納することにより達成されてもよい。コンテキストには、例えば、異なるオペレーティングシステム及びコンテキストの変換を必要としない言語を使用する別の制御器又はクライアント装置に転送可能なＸＭＬによるコンテキストがある。

【０１０４】

付録２には示さないが、タスクマネージャ１４は、タスクの履歴及びタスクの現在の状態に関連付けられたデータがＸＭＬコンテキストに格納されるように、タスクを構成してもよい。これにより、サービスが実行されると、タスクは、サービスの実行の結果として取得されるデータと共に格納され、ユーザが必要とする場合には、次のサービスの実行を制御できる別の制御器又はクライアント装置に転送される。

【０１０５】

上述の実施形態において、サービスディスカバラは、プラグインアーキテクチャを使用する。このことは、上述の理由により、利点を有するが、サービスディスカバラ１２は、あるプロトコルで動作するサービスのみを発見できる専用のサービスディスカバラとして

実現されてもよい。同様に、上述の実施形態において、タスクマネージャは、必要に応じて置き換え又はグレードアップされるモジュールとして実現されてもよい。これは、上述の通り利点を有するが、タスクマネージャを演算制御器内に実装することも可能である。しかし、タスクマネージャのグレードアップ又は置き換えは、更に困難になる。

付録1

サービス記述の例

```
<?xml version="1.0" encoding="UTF-8" ?>
- <CIAServiceGroup>
- <CIAService name="WebGalleryService" type="Synthesize"
    GUID="4f1d0d:fbee5bda43:-7ff2">

    <URI>class:WebGalleryService.jar:com.canon.cre.cia.services.webgall
    ery.WebGalleryService</URI>
- <Description>
  <Icon type="mime" format="image/png">iVBOR... QmCC</Icon>
  <Info lang="en" name="Web Gallery">This Service creates a web gallery
    from a series of images</Info>
  <Info lang="fr"
    name="Galerie Web">Ce service crée un galerie web d'une serie
  d'images</Info>
    </Description>
- <Inputs>
  <Input id="image" type="mime" format="image/jpeg" multiple="true"
    required="true" />
    </Inputs>
- <Outputs>
  <Output id="html" type="mime" format="text/html" multiple="true" />
  <Output id="fullImage" type="mime" format="image/jpeg" multiple="true"
    />
  <Output id="thumbImage" type="mime" format="image/jpeg"
    multiple="true" />
    </Outputs>
- <Parameters>
  <Parameter id="title" type="simple" format="string" />
  <Parameter id="thumbsPerRow" type="simple" format="int" />
  <Parameter id="thumbWidth" type="simple" format="int" />
  <Parameter id="thumbHeight" type="simple" format="int" />
  <Parameter id="webFileName" type="simple" format="string" />
  <Parameter id="webImagesDirectory" type="simple" format="string" />
  <Parameter id="webThumbPrefix" type="simple" format="string" />
    </Parameters>
    </CIAService>
- <CIAUIService name="WebGalleryUI" type="SynthesizeUI"
    GUID="4f1d0d:fbee5bda43:-7ff1">
  <Service name="WebGalleryService" />
  <UILanguage name="java" />
    </CIAUIService>
    </CIAServiceGroup>
```

付録２

タスク記述の例

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
UTF-8 Chars: □□□
  -->
- <CIATask name="SendOCR">
  <Device name="iR" />
  <Output uri="address" id="address" type="simple" format="string" />
- <Description>
  <Info lang="en" name="AddressBook" />
    </Description>
- <Description id="email">
  <Icon uri="scan.png" />
  <Info lang="en" name="Scanned Document">Scan a document and sends
    via email</Info>
   <Info lang="fr"
    name="Email le document">Scanner un document et envoyer par email</Info>
    </Description>
- <Description id="scan">
  <Icon uri="email.png" />
  <Info lang="en" name="Email Document">Scan a document and sends via
    email</Info>
   <Info lang="fr" name="Email le document">Scanner un document et envoyer
    par email</Info>
     </Description>
- <Service id="address" name="OpenAddressBook" type="ContactOutput">
  <Output id="address" required="true" multiple="true"
    targetServiceID="service0" targetInputID="address" />
    </Service>
- <Service id="service2" name="ScannerService" type="Scanner">
  <Parameter id="resolution" type="simple" format="int">300</Parameter>
  <Output id="image" required="true" multiple="true"
    targetServiceID="convert" targetInputID="inputData" />
  <Output id="image" required="true" multiple="true"
    targetServiceID="service1" targetInputID="image" />
    </Service>
- <Service id="convert" name="ConvertToPDFService" type="Convert">
  <Input id="inputData" required="true" multiple="false" />
  <Output id="outputData" required="true" multiple="true"
    targetServiceID="service0" targetInputID="attachments" />
    </Service>
- <Service id="service0" name="EmailService" type="Email">
  <Parameter id="subject" type="simple" format="string">Document from
    %fullname%</Parameter>
  <Input id="subject" required="false" multiple="false" />
```

```
    <Input id="body" required="false" multiple="false" />
    <Input id="address" required="true" multiple="true" />
    <Input id="attachments" required="false" multiple="true" />
      </Service>
  - <Service id="service1" name="OcrService" type="Ocr">
    <Input id="image" required="true" multiple="true" />
    <Output id="text" required="true" multiple="true"
      targetServiceID="service0" targetInputID="body" />
      </Service>
  - <UI name="OCREmailUI">
    <Service id="service0" />
    <Service id="service1" />
    <Service id="convert" />
      </UI>
  - <UI name="AddressBookUI">
    <Service id="address" />
      </UI>
        </CIATask>
```

【図面の簡単な説明】

【０１０６】

【図１】タスクの実行を容易にする本発明を実現する制御器を含むネットワークシステムの機能ブロック図である。

【図２】本発明を実現する制御器の更に詳細な機能ブロック図である。

【図２ａ】サービス記述を概略的に示す図である。

【図３】図１及び図２に示す制御器を提供するように構成された演算装置の一例を示す機能ブロック図である。

【図４】図１及び図２に示す制御器のタスクマネージャがタスクの実行を管理する１つの方法を示すフローチャートである。

【図５】新しいサービスを発見する制御器の動作を示すフローチャートである。

【図６】サービスにアクセスする制御器の動作を示すフローチャートである。

【図７】サービスの要求に対する制御器の動作を示すフローチャートである。

【図８】タスクマネージャがタスクの実行を管理する別の方法を示すフローチャートである。

【図１】

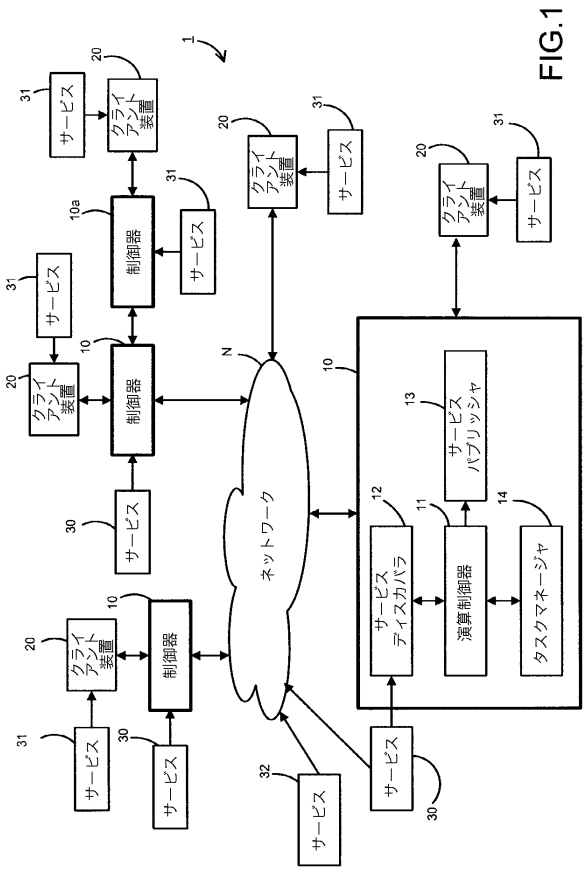


FIG.1

【図２】

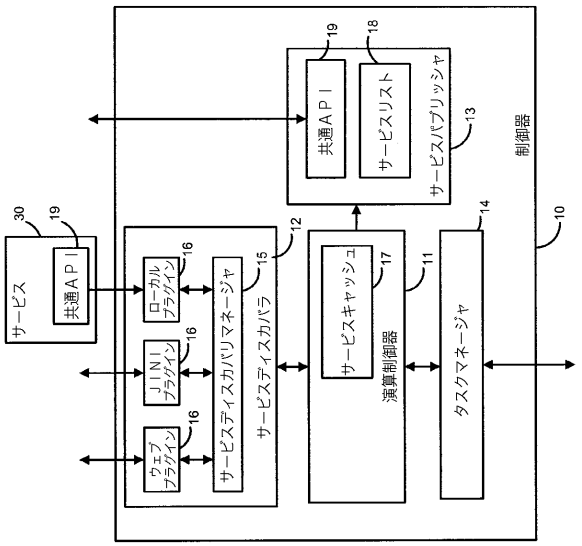


FIG.2

【図２ａ】

100

サービス	経路情報
サービス名	
ID	
種類	
分類	
入力	
出力	
パラメータ	
機能	

FIG.2a

【図３】

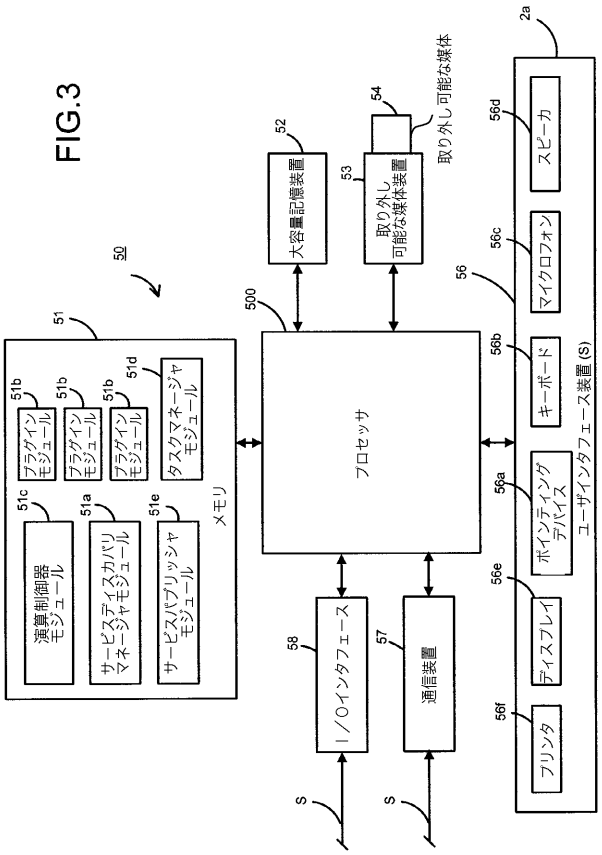


FIG.3

【図４】

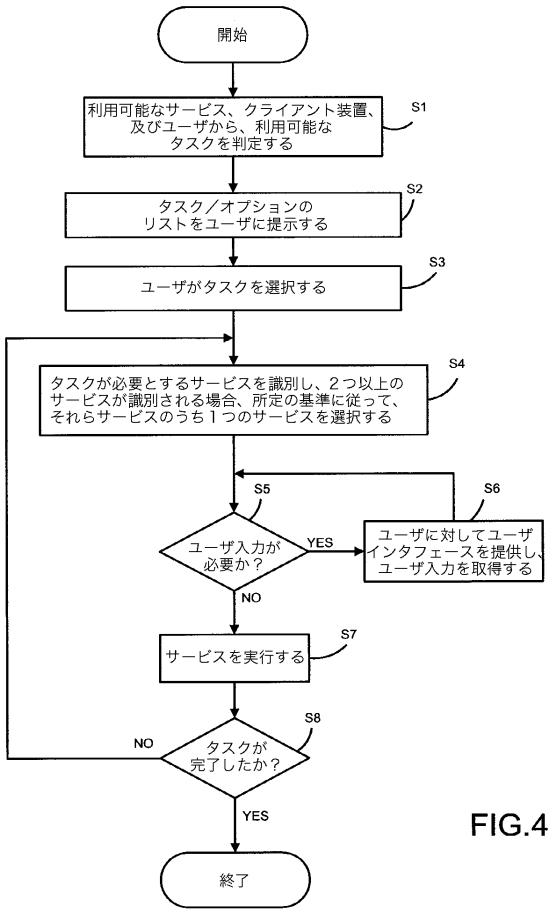


FIG.4

【図５】

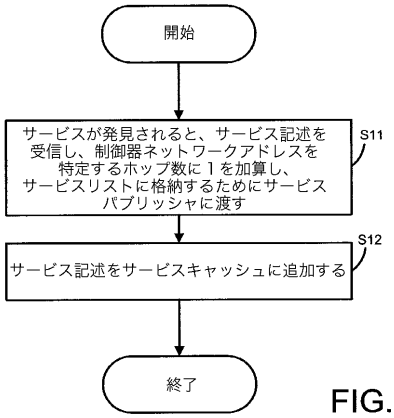


FIG.5

【図６】

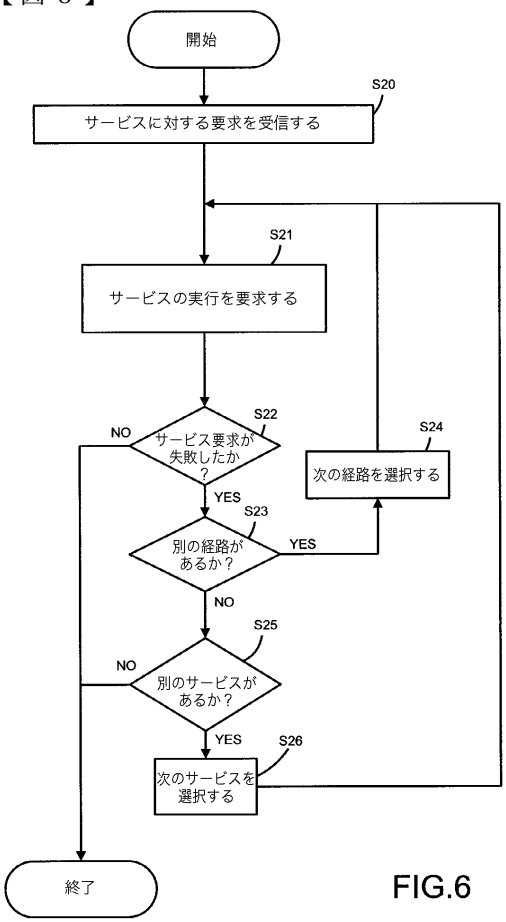


FIG.6

【図７】

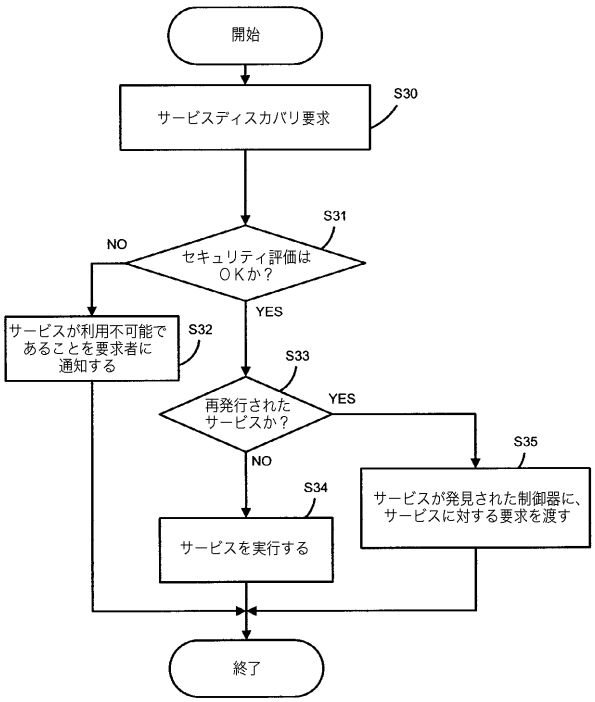


FIG.7

【図８】

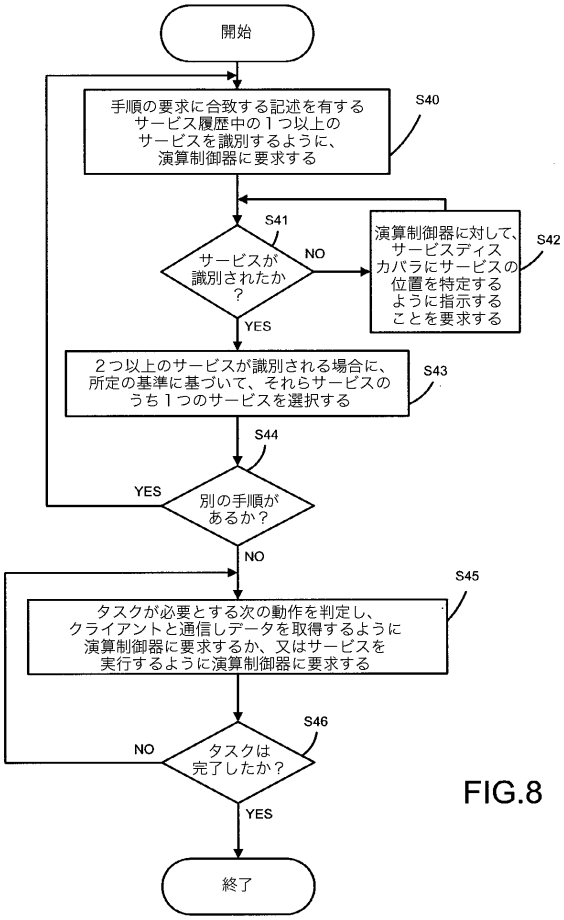

開始
手順の要求に合致する記述を有する
サービス履歴中の１つ以上の
サービスを識別するように、
演算制御器に要求する
S40
S41
サービスが
識別されたか
？
NO
演算制御器に対して、
サービスディス
カバラにサービスの
位置を特定する
ように指示する
ことを要求する
S42
YES
２つ以上のサービスが識別される場合に、
所定の基準に基づいて、それらサービスの
うち１つのサービスを選択する
S43
S44
YES
別の手順が
あるか？
NO
S45
タスクが必要とする次の動作を判定し、
クライアントと通信しデータを取得するように
演算制御器に要求するか、又はサービスを
実行するように演算制御器に要求する
S46
NO
タスクは
完了したか？
YES
終了

FIG.8

フロントページの続き

(72)発明者　ピーター　マイケル　アールデイ
イギリス国　アールジー１２　２エックスエイチ，　バークシャー，　ブラックネル，　ロンドン　ロード，　ザ　ブラカンズ　キヤノン　リサーチ　センター　ヨーロッパ　リミテッド　内
(72)発明者　ジョージオス・ヴァ　シロプロス
イギリス国　アールジー１２　２エックスエイチ，　バークシャー，　ブラックネル，　ロンドン　ロード，　ザ　ブラカンズ　キヤノン　リサーチ　センター　ヨーロッパ　リミテッド　内
(72)発明者　アンドリュー・ミリン
イギリス国　アールジー１２　２エックスエイチ，　バークシャー，　ブラックネル，　ロンドン　ロード，　ザ　ブラカンズ　キヤノン　リサーチ　センター　ヨーロッパ　リミテッド　内
(72)発明者　ウイルソン　シーエン　チュウ　チウ
イギリス国　アールジー１２　２エックスエイチ，　バークシャー，　ブラックネル，　ロンドン　ロード，　ザ　ブラカンズ　キヤノン　リサーチ　センター　ヨーロッパ　リミテッド　内

【外国語明細書】
1.Title of Invention
APPARATUS FOR AND A METHOD OF FACILITATING THE CARRYING OUT OF A TASK

2.Field of the Invention
This invention relates to apparatus for and a method of facilitating the carrying out of a task. In particular, this invention relates to apparatus for and a method of facilitating the carrying out of a task by a processor-controlled machine in a distributed system or network in which a number of processor-controlled machines such as personal computers, servers, digital cameras, photocopiers and the like have or have access to services that may be required for the carrying out of a task.

3.Background
In order to complete a complex task on a network, a user may have to make use of several different applications to perform different operations. Thus, for example, if a user wishes to email a copy of a paper document so that the recipient can subsequently edit the document using a word processing application, then, generally, the user will need to use a scanner application to convert the paper document into electronic form, then use an optical character recognition application to convert the electronic data into electronic character data and then use an emailing application to email the final electronic document to the desired recipient.

A task-based approach to such complex operations simplifies the number of individual operations that a user has to perform. In a task-based approach, a controller forming part of the network (for example a server with which the user's processor-controlled machine can communicate or the user's processor-controlled machine itself) identifies the task to be carried out from data provided by the user and then either selects a predefined task consisting of the required services or assembles a new task consisting of the required services. Thereafter, the user simply needs to provide user input data when prompted by the controller to enable the carrying out of the task.

Thus, in the example given above, where the user advises the controller that he wishes to email a hard copy document in editable form, then the controller will either access a predetermined task for carrying out that sequence of steps or will assemble a new task to carry out that predetermined sequence of steps. In either case, the task in this example will consist of a set of services including a scanning service, an optical character recognition service and an emailing service.

The number of tasks that can be carried out is limited by the services that the controller can access and, although a controller may be able to access services remotely over the network, it is difficult if not impossible for a controller to access services that are only coupled to the network via other controllers.

4.Detailed Description of Invention
In one aspect, the present invention provides a controller or apparatus for enabling the carrying out of task, wherein the apparatus is connectable to a network

and comprises a service discover operable to discover services over the network, a task manager operable to control the carrying out of tasks using services accessible by the apparatus and a service publisher for publishing descriptions of services accessible to or discovered by the apparatus so that other controllers coupled to the network can see the services published by the controller even though at least some of those services may be connected to the network only via the controller.

In one aspect, the present invention provides a controller for facilitating a carrying out of a task, wherein the controller is connectable to a network and looks to other controllers like a service thereby allowing other controllers to discover and access the services via that controller.

This enables a controller to access services that it may not be able to see directly and as a result allows more or more flexible tasks to be offered to the user.

In an embodiment, a controller is configured to enable the list of services that are passed on to another controller to be subject to security restrictions. For example,the controller may be configured to restrict the list of services that are passed on to another controller on the basis of the identity of the user that is trying to access them and/or on security restrictions that are placed on the service by the at least one controller in the route from the location of actual service to the requesting controller.

A controller may be configured to provide different security restrictions dependent upon whether the user is local to the controller or whether a service is being requested by a remote controller.
A controller may be configured to increment a hop count each time a description of a service is passed to that controller so that, for example, where a controller receives the same service description more than once, the controller can determine the most direct route to the actual location of that service.

A controller may be configured to add descriptions of services that have been discovered more than once to a cache so that, if a route to a particular service fails, then that same service may be tried again via a different route.

In an embodiment, the services available for the carrying out of a task will generally include input services, output services, and processing services. In addition, the available services may include user interface services. This avoids the possibility of the services available to a user being restricted by the user interface of the device via which the user accesses the controller. Rather, by providing the user interfaces as services, then the user interface required by a task or indeed user interfaces required by different services within a task can be loaded on to the user's device as and when required so that dedicated user interfaces can be provided for different services within a task. This has the advantage that the user interface presented to the user is not cluttered with information which is particularly important where the size of the display on the user's device is relatively small. In addition, providing the user interfaces as services enables user interfaces for existing services and tasks to be updated a

s and when required.

Different protocols may be required for discovering different services. In an embodiment, the services use a common API (application programming interface) and the service discoverer is configured with a plug-in architecture so that the service discoverer has one or more plug-in service discovery modules, each of which is designed to discover services that use a particular protocol or protocols. For example, one service discovery plug-in module may be designed to discover Web services, another service discovery plug-in module may be designed to discover JINI services and another service discovery plug-in module may be designed to discover local services, that is services directly connected to the controller. The use of a plug-in architecture means that a controller can be upgraded or updated when new protocols come into use simply by adding a plug-in configured to operate with that protocol. Thus, when a new protocol becomes popular, it can easily be supported and deployed without changing the rest of the controller. In addition, controllers of smaller devices (such as digital cameras) can just use the plug-in or plug-ins that they require. The use of such plug-ins and a common API for services means that a service which uses a protocol for which a plug-in has already been developed can be used straight away without any further adaptation and, if a new protocol is developed or used, then only one plug-in needs to be developed to handle services using that protocol.

In a task based system, some services may require certain parameters to be set by a user. However, a user may not always know the best value for a parameter. Therefore, it is often desirable for a default parameter to be set.

In an embodiment, default parameters are provided on different levels. Thus, a service may itself have a default parameter and this default parameter may be overridden by a default task parameter, so that when a particular service is running in a particular task, the default parameter is appropriate to that task. Thus, for example, a scanning service may have its own default scanning resolution parameter which, for example, represents an average resolution while a task to scan and email a photograph may have a different resolution default parameter that overrides the service default parameter so that, when scanning a photograph for emailing, the scanner scans at a relatively low resolution to reduce the size of the image file while where the scanning services is used in a task in which a document is to be subject to optical character recognition, for example the above task to scan and email in editable form a hard copy document, then the task may define a high resolution default scanning parameter to enable accurate optical character recognition. In addition, a user interface service may enable a user to change or customize the service level default parameter or the task level default parameter in accordance with their requirements.

The device at which a user commences a task may not necessarily be the best device to enable completion of the entire task. Thus, for example, in the case of the above task to scan and email in editable form a hard copy document, the user may wish to use a networked photocopier to perform the scanning but then move to a personal computer to enable access to a keyboard to enter information. In addition, some services may require more processing power than other services and it may speed up completion of a task if a high processing power service is compl

eted by a device having a lot of processing capacity, for example it may be more efficient for a particular service to be carried out by a networked personal computer rather than a networked digital camera.

In an embodiment, a task is designed so that it can be stored in an incomplete form and transfer to another, for example more powerful, controller to enable completion of the next portion of the task. In an embodiment, the current state of a task and the data associated with it is stored in a data structure known as a context, for example an XML based context, which can be passed between controllers allowing a user to access the task from different client devices. This means that a user can commence a task at one client device and move to another that may be more suitable for continuation of the task with, in each case, the controller controlling the coordination and processing of the task so as to enable the task state to be stored for later retrieval or passed to another controller for continued processing.

Referring now to the drawings, Figure 1 shows a functional block diagram of a network system 1 embodying the invention.

The network system 1 comprises a number of controllers 10 in the form of personal computers, servers, other processor device or the like each connected to one or more client devices 20 by which a user can input instructions and/or data for instructing the controller to carry out a task using services accessible by the controller. As shown in Figure 1, a controller 10 may have local services 30. In addition, services 32 may be directly connected to the network N and services 31 may be directly connected to a client device 20.

The controllers 10 may be directly coupled to the network N or via one or more other controllers 10. In the example shown in Figure 1, three controllers 10 are directly coupled to the network and a further controller 10a is indirectly coupled to the network via another one of the controllers. It will, of course, be appreciated that the actual network configuration will depend upon the particular circumstances and that, for example, fewer or more than three controllers may be directly coupled to the network. Also, the network may comprise further controllers 10a that are only coupled to the network indirectly via another controller. In addition, the network system may also include further controllers that are coupled to the network via two or more other controllers.

Although not shown in Figure 1, each client device 20 may incorporate or be integrated with a controller 10. In addition, a client device 20 may communicate with a controller 10 via the network N.

The client devices 20 are processor-controlled machines provided with network connectivity and each having at least one user interface device for enabling a user to interface with the client device. Examples of client devices are networkable devices such as personal computers, photocopiers, facsimile machines, digital cameras, scanners and other items of office equipment where the network is intended for use in an office environment.

It will, of course, be appreciated that the network may be intended for use in o

ther than an office environment. Thus, for example, where the network is in a h ome environment, then the client devices may include personal computers and othe r networkable processor-controlled machines such as, for example, televisions, v ideo recorders, DVD players, and other processor-controlled machines or items of equipment that may be found within the home. Similarly, the network system may be used in an industrial environment where one or more of the client devices ma y comprise processor-controlled manufacturing plant or machine tools.

The network N may comprise one or more different types of network. For example, the network N may comprise at least one of a local area network 11 (LAN), a wid e area network (WAN), an intranet and the Internet.

Figure 1 illustrates the main functional components of one of the controllers 10 . It will be appreciated that each of the other controllers 10 and 10a has the same main functional components.

As shown in Figure 1, a controller 10 has an operations controller 11 that contr ols overall operation of the controller and a task manager or task resolution mo dule 14 for controlling or managing the carrying out of tasks. In addition, eac h controller 10 has a service discoverer 12 for discovering both local services 30 and services on the network N and a service publisher 13 that, as will be des cribed below, is configured to make services discovered by the controller 10 dis coverable by other controllers 10. That is, the service publisher 13 is configu red to advertise or republish discovered services so that they are accessible by other controllers 10 over the network N.

The operations controller 11 is also configured to communicate with the client d evice or devices 20 connected to the controller 10 and to communicate with the s ervices required during the carrying out of a task and to coordinate the operati on of the next step in a task as requested by the task manager 14. In addition the operations controller 11 is configured to pass on calls made to services pub lished by the service publisher 13, that is services that are not directly avail able to that controller 10.

The task manager 14 is configured to determine which task(s) can be run using th e currently available client device(s) 20 and services and to determine the next action required, for example whether data is required from a user or whether a service should be run.

The task manager 14 is, in this embodiment, configured as a separate module so t hat it can be swapped in and out easily so that a controller 10 has a task manag er that is appropriate to the client device 20, and to enable easy modification of the system.

As mentioned above, each of the controllers 10 may have access to a number of lo cal services 30. In addition, each of the client devices 20 may have direct acc ess to one or more services 31 and one or more services 32 may be directly coupl ed to the network N to enable access by all of the controllers 10.

The services 30, 31 and 32 are stored in appropriate memory. The services 30 ma

y be stored in a central data storage of the network system 1 while the services 30 and 31 may be stored in any appropriate memory available at the controller 10 or client device 20, for example a mass storage device such as a hard disc drive in the case of a computer, or on a memory card in the case of a client device such as a digital camera.

Generally speaking in this embodiment there are four types of services:

1) input services that provide input data to a task, an example being a scanning service;
2) output services that output data from a task such as, for example, printing services and email services;
3) processing services that perform a process on or modify input data such as, for example, image manipulation services, optical character recognition services and so on; and
4) user interface services that provide user interfaces for loading onto client devices.

A service may itself consist of a set of sub-services so that, for example, an address book service may consist of the sub-services "open address book" and "save address book" plus also possibly an address book user interface sub-service.

Each service uses the same common API (Application Programming Interface) which implements four methods:

1. Get description
2. Get status
3. Run service
4. Get user interface(s)

The common API therefore enables a controller 10 to discover services and to obtain their description and status and, when the task manager 14 indicates that a service is required, to get the associated user interface or interfaces and to run the service.

Figure 2a shows a very diagrammatic representation 100 of a service description. The service description includes the following components:
1. Service name
2. ID
3. Type
4. Inputs
5. Outputs
6. Parameters
7. Capabilities

The service description optionally also includes a classification component.

The service name is unique to the service so that if, for example, there is more than one controller 10 or client device 20 on the network N that runs the same service, they will be allocated the same service name while the ID is unique to

the particular service and controller or client device that actually runs that service and may be in the form of a serial number or other unique code.

The type component defines the type of service and is descriptive of the operation or procedure that the service performs such as scan, email, OCR, print, image manipulation, and so on.

The optional classification component represents a lower level description of the service and identifies features available via that service. Thus, for example, in a case of a service of the type "image manipulation", the classification will specify the particular nature of the image manipulation provided by the service, examples of possible classifications for an image manipulation service are: re-size, red eye removal, rotate, make panorama and so on.

The input component defines the type or types of data that can be passed to the service and the format or formats required for that data while the output component defines the data type or types that the service provides and the format or formats that that output data has.

The parameter component comprises a value or values that can be passed to the service and that affect(s) what the service does with data. The service description may contain a default value for one or more parameters. An example of a parameter for a print service would be, for example, a value for the number of copies.

The capability component provides a description of what the service is capable of. Thus, for example, the capability component may define the maximum resolution in dots per inch of a scanning or printing service.

Each input, output and parameter has an ID that describes the nature of the data (for example image or document), a type which identifies the family that the data belongs to and a format that identifies the exact format that the data takes. Thus, for example, if the type is "MIME" then the format may be image/png or application/word whereas if the type is "simple" then the format may be "string" or "int".

In addition, inputs, outputs and parameters may specify a constraint or limit on those properties, for a maximum size of input data or a maximum or minimum resolution may be specified.

The service description thus provides a controller 10 with the information necessary to determine whether the service is suitable for and can carry out an operation required for a particular task.

Figure 2 shows a more detailed functional block diagram of one example of a controller 10.

As shown in Figure 2, the service discoverer 12 comprises a service discovery manager 15 configured to receive a number of plug-ins 16 (three in the example shown). Each of the plug-ins 16 is configured to enable discovery of services oper

ating in accordance with certain protocols. The plug-ins 16 may comprise, as sh own, a plug-in configured to discover Web services, for example operating in acc ordance with the UDDI (Universal Description Discovery and Integration) protocol and the SOAP (Simple Object Access Protocol) communications protocol, a plug-in configured to discover JINI services and a plug-in configured to discover local services. Figure 2 shows the local plug-in 16 having discovered a local servic e 30. The fact that the services use or implement a common API is illustrated d iagrammatically in Figure 2 by using a block 19 to represent the common API.

The use of a plug-in architecture means that, as new discovery systems and proto cols are developed, new plug-ins 16 can be designed to enable discovery of servi ces using those protocols. In addition, a controller 10 need only use those plu g-ins required by the client(s) 20 with which it communicates. Thus, for exampl e, if a controller 10 is only communicating with a dedicated processor-controlle d machine such as a digital camera, then only those plug-ins commensurate with t he services that a digital camera can use may be provided.

In addition to the plug-in architecture, the services useable by the network sys tem are all configured to use the same common API. The use of a common API enab les a new service to be used straight away, provided that a plug-in exists for t he protocol(s) required by that service.

As shown in Figure 2, the operations controller 11 maintains a services cache 17 in which the service descriptions of discovered services are held. In addition , the service publisher 13 stores a services list 18 containing the service desc riptions of all of the services that have been discovered by the controller 10. The service publisher 13 is also configured to communicate with the network N a nd the directly connected client device(s) 20 using the common API (again repres ented diagrammatically by a block 19) so that the other controllers 10 on the ne twork N and the connected client device(s) 20 can see the controller as a servic e.

The controllers 10 may be implemented by programming computing apparatus such as a personal computer or server or other processor device. Figure 3 shows a funct ional block diagram of one example of computing apparatus 50 that may be program med by program instructions to provide a controller 10.

As shown in Figure 3, the computing apparatus 50 comprises a processor 500 with associated memory 51 in the form of ROM and/or RAM and a mass storage device 52 such as a hard disk drive. The computing apparatus also comprises a removable m edium device 53 for receiving a removable medium 54 such as, for example, a CDRO M, DVD, floppy disk or the like. In addition, the computing apparatus comprises a number of user interface devices 56. As shown these include a pointing devic e 56a such as a mouse, a keyboard 56b, a microphone 56c, a loudspeaker 56d, a di splay 56e and a printer 56f. The computing apparatus also comprises one or more communications devices 57 such as a network card and/or a MODEM for enabling co mmunication over the network N and one or more input/output interface(s) 58 for enabling communication with external devices such as the client device 20 and th e services 30 directly linked to the controller 10.

The computing apparatus is programmed by at least one of:

program instructions pre-stored in the memory 51 or in the mass storage device 52;

program instructions provided as a signal via a communications device 57 or an I/O interface 58.

program instructions downloaded from a removable medium 54 received via the removable medium device 53 or from a portable data storage device connectable to the computing apparatus via an I/O interface such as a USB port; and

program instructions directly entered by the user using the keyboard 56b.

Figure 3 shows the computing apparatus as having been programmed so that the memory 51 contains a service discovery manager module 51a for implementing the service discovery manager 15, plug-in modules 51b for implementing the plug-ins 16, an operations controller module 51c for implementing the operations controller 11, a task manager module 51d for implementing the task manager 14 and, a service publisher module 51e for implementing the service publisher 13.

It will be appreciated the client devices 20 will, being processor controlled machines, also have a processor with associated memory, I/O interfaces and some form of user interface device(s) which, in the case of a personal computer, will be similar to that shown in Figure 3 and in the case of a digital camera will be more limited generally consisting of a small display and a user control that enables selection of items from a displayed menu.

The manner in which a controller 10 enables the carrying out of a task based on a number of services will now be described.

A task consists of one or more services, depending upon the particular task. Common tasks or tasks that have been or are likely to frequently used may be stored by the task manager 14. Other tasks may be assembled by the task manager 14 from available services in accordance with input information received from the user.

In order for a task to be carried out it has, of course, first to be initiated by a user. In order to facilitate this, a client device 20 may be provided with a basic task user interface that prompts the user with a question such as:

What do you want to do?

and may present the user with a number of options defining currently available tasks. The nature of the tasks available will generally depend upon the particular client device. As an example, if the client device is a networked photocopier, then task options may include:

Scan
Scan and email

Scan, OCR and email

As another possibility, the basic task user interface may enable the user to define a task by selecting options from a displayed menu, for example if the user wishes to scan a hard copy document and email it in editable form the user may select displayed options "scan", "optical character recognition", and "email" from a menu.

Figure 4 is a flowchart illustrating steps carried out by the task manager.

As set out above, the controller 10 stores the description for discovered services. At S1 in Figure 4, the task manager 14 determines from the available discovered services, the client device and the user which tasks are available to that client device and user.

Then, at S2, the task manager causes the client device 20 to present the user with a list of tasks/options available to that user, one example of an available task may be "scan a hard copy document, perform optical character recognition and then email the document in an editable form".

When, at S3, the user selects a task from the list presented to him at the client device then, at S4, the task manager identifies a first service required by the task by checking the type components of the stored service descriptions to determine which of the type components match the tasks requirements and, of those, which services have input, output, capability and classification parameters acceptable for the task.

It is possible that more than one service may match the requirement. If so, then at S4, the task manager selects one of those matching services in accordance with predetermined criteria as will be described below.

Once a service has been selected for the first procedure or step of the task then, at S5, the task manager 14 determines whether user input is required and, if so, causes the client device to present the appropriate user interface to the user to get the required user input at S6.

If the answer at S5 is no or if user input has already been obtained at step S6 then, at S7, the task manager requests the operations controller 11 to cause the required service to be run.

Then, at S8, the task manager checks to see whether the task has been completed, that is whether the task involves a further procedure that requires another service.

If the task manager determines that the task involves a further procedure that requires another service then steps S4 to S7 are repeated and at S8 the task manager again checks whether there are further procedures that require services to be run in order for the task to be completed. The procedure ends once the task manager determines at S8 that there are no further procedures requiring services to be run.

Figure 5 shows steps carried out when a service is discovered.

When the operations controller 11 starts the service discoverer 12 it registers a Discovery Listener with the service discovery manager 15. When the service discoverer 12 locates a plug-in 16, it starts the plug-in 16 and then registers the service discovery manager 15 with the plug-in 16. When a service 30,31,32 is discovered by a plug-in 16, the plug-in 16 notifies the service discovery manager 15 which notifies the operations controller 11.

Upon discovery of a new service then, at S11 in Figure 5, the operations controller 11 receives the service description, updates route information associated with the service description (for example by adding its controller network address to route address and by adding one to a hop count, where a "hop" is a step or passage from one controller to the next on the network) and, having stored the service description with its associated route information in its service cache 17, passes the service description with its associated route information to the service publisher 13 for storage in the services list 18. As another possibility, the hop count may be updated by the service publisher 13.

Because the service publisher 13 of each controller 10 republishes (that is makes accessible to other controllers on the network N) discovered services by using or implementing the common API 19, the same service may be discovered by a controller 10 via a number of different routes. In the event that a service is discovered more than once, then the service description is still added to the service cache 17 step S12 so that, if a subsequent request to run that service fails, the operations controller 11 can try and access the service via the other route or one of the other routes.

Figure 6 shows a flowchart for illustrating operations carried out by the operations controller 11 when the task manager 14 issues a request for a service to be run at S20. Thus, at S21, the operations controller 11 implements the run service method of the service description of the selected service at S21 to cause the service to be run. If the service is not a local service, then the controller 10 causes the request for the service to be run to be passed on to the controller from which the service was discovered and, if that controller is not the controller that actually runs that particular service, that controller will, in turn, pass the request for the service to be run on to the controller from which it discovered the service and so on until the request for the service to be run reaches the controller that actually runs that particular service. The controller that actually runs that particular service will then invoke the service and return the results to the requesting controller 10.

If, at S22, the request for a service to run fails, for example because one or more of the network addresses along the route path is inoperable or the service is currently busy, then at S23, the operations controller 11 determines whether there is another route for the same service and, if so, selects that other route at S24 and repeats steps S21 to S23.

If the answer at S23 is no, that is the service has failed and there is no other

route that has not yet been explored, then at S25, the operations controller checks whether there is another acceptable service and, if so, selects that other service at step S26 and repeats steps S21 to S26.

Where more than one service is available for carrying out a particular procedure, then as set out at S4 in Figure 4, one of the services may be selected on the basis of predetermined criteria. These criteria may include at least one of:

1. the number of hops required to reach the controller that runs the service, on the grounds that this may determine the reliability of the service and the time taken to access the service;
2. a rating system;
3. required processing time;
4. current status of the service;
5. costs;
6. quality;
7. recommendations from users on the system.

One or more of the above preferences may be determined by information input by the user via an appropriate user interface.

In addition to the above, dependent upon the client device or devices 20 that the controller 10 services, the controller 10 may be specialised so that it filters through only particular services. This may be achieved by selection of the plug-ins so that only services complying with particular protocols are discovered or may be achieved by the operations controller 11 filtering through only services whose service descriptions match certain requirements. For example where the client device is a digital camera, then the controller 10 may be configured to filter through only services that are applicable to a digital camera so excluding, for example, word processing services.

The operations controller 11 may also have security settings that determine whether or not a service will be passed on to a requesting controller. Figure 7 shows a flowchart illustrating one way in which this may be implemented. Thus, when at S30, a controller 10 receives a service discovery request, that request will be accompanied by data that identifies the requesting controller and may also identify the requesting user.

At S31, the operations controller 11 determines whether the security rating for the requesting controller or task manager and/or user enables them to have access to the requested service and, if not, advises the requesting service discoverer 12 that the service is not available.

An operations controller 11 may have different security ratings for local and remote users.

In the event that the operations controller 11 determines that the requesting controller and/or user has/have an appropriate security rating, then at S33, the operations controller checks whether the service is a republished service and, if not, causes the service to be run at S34. If, however, the service is a republ

ished service then at S35, the operations controller passes the request for the service on to the controller from which the service was discovered. That contro ller then repeats the procedure shown in Figure 7 so that the request is passed from controller to controller until it reaches the controller that runs the requ ested service.

As shown in Figure 7, the controller is arranged to check the security rating be fore determining whether or not a service is a republished service. This enable s each controller to have its own security ratings for remote services. As anot her possibility, steps S33 and S31 may be reversed so that the operations contro ller 11 first checks whether the service is a republished service and, if so, pa sses the request onto the controller from which the service was discovered witho ut checking the security ratings. This would mean that whether or not a particu lar controller and/or user is given access to a service will be determined solel y by the controller that initially discovered that service.

Figure 4 illustrates one way of carrying out a task. Figure 8 illustrates anoth er way in which a task may be carried out once the user has selected a task or has indicated what they require of the task for example scan a hard copy documen t, perform optical character recognition and then email the document in an edita ble form.

As set out above, the controller 10 stores the descriptions for discovered servi ces.

At S40 in Figure 8 the task manager 14 requests the operations controller 11 to identify from the description stored in the service cache 17 a service or servic es matching the requirements for a first procedure or step of the task.

The task manager 11 then checks the type components of the stored service descri ptions to determine if any of the type components match the task manager's requi rements and, if so, determines whether the input, output, capability and classif ication parameters in that service description are acceptable for the task in ha nd.

If, at S41, the operations controller 11 reports that no appropriate services ha ve been identified, then at S42, the task manager 14 requests the operations con troller 11 to instruct the service discovered 12 to locate a service.

Moe than one service may match the task requirements. Accordingly, at S43, the task manager 14 determines whether more than one service has been identified and , if so, selects on of the services on the basis of predetermined criteria as wi ll be discussed below.

Once a service has been selected for the first procedure or step of the task the n, at S44, the task manager 14 determines whether the task involves a further pr ocedure that requires another service and, if the answer is yes, repeats steps S 40 to S43.

Once services have been discovered for all of the procedures required to complet

e the task then at S45 the task manager 14 determines the next action or procedu re required by the task and requests the operations controller 11 to act accordi ngly. Thus, if the task manager determines at S45 that user input is required a nd that this does not include a new user interface then the task manager request s the operations controller to communicate with the client device 20 to prompt t he user to input the necessary data. Otherwise the task manager determines the next action required is for a service to be run, then the task manager requests the operations controller 11 to cause the required service to be run. The requi red service may in some cases include a sub-service such as a user interface ser vice and in those cases the operations controller 11 will first download the req uired user interface to the client device 20.

Step S45 is repeated until the task manager determines at S46 that the task has been completed, that is all of the services required to complete the task have b een run.

As described above, each of the controllers republishes discovered services usin g or implementing the common API so that, as far as other controllers are concer ned, each controller also looks like a service. The services available may incl ude input services, output services, processing services and user interface serv ices. The provision of user interfaces as services means that the user interfac e can be loaded onto the client device and run as and when required so that dedi cated user input services can be provided for particular services and, in additi on, the user interface for a particular client device may be easily updated and/ or modified.

As described above, a service may include default parameters for the operation p rovided by this service. For example, where the service is a scanning service, then a default scanning resolution may be defined. Similarly, where the service is a printing service, then a default printing characteristic such as black and white, one-sided and so on may be defined. In some tasks, specific parameters may be required of a service. Thus, for example, in the case of a task to scan and then email in editable form a hard copy document, the optical character reco gnition software may require a scanning resolution higher than the normal defaul t scanning resolution of the scanning services. In order to cope with such occu rrences, the task manager 14 may be configured to provide the task itself with o ne or more default parameters and the task may be configured such that the task default parameters override service parameters. Taking as an example, the task mentioned above, this would enable the task manager 14 to define for the task a default scanning resolution parameter higher than the default resolution scannin g parameter of the scanning service. In contrast, where the task to scan and em ail a photograph the task manager 14 may define a low resolution scanning parame ter that overrides the scanning services default parameter so that an image file that is not too large, that is appropriate for emailing, is produced by the sca nning service. In addition, a user interface service for a task or a service wi thin a task may enable a user to change or customize the service level default p arameter or the task level default parameter in accordance with the particular u sers requirements.

As so far described above, once a task has been defined and the necessary servic

es assembled, the operations controller 11 and the task manager 14 of the contro ller 10 at which the task originated coordinate carrying out of the task with th e services required by the task being run by the controllers which initially dis covered those services.

This may, however not necessarily always be convenient for a user, whose local c ontroller may change as he moves to different physical locations serviced by the network. In addition, certain client devices may be more convenient when carry ing out certain services within a task. Thus, for example, in the case of the a bove task to scan and email in editable form a hard copy document, the user may wish to use a networked photocopier to perform the scanning because this is quic k than using a scanner attached to a personal computer but may then wish to move to a personal computer to enable access to a keyboard to enter an email message . In addition, some services may require more processing power than other servi ces and it may speed up completion of a task if a high processing power service is completed by a device having a lot of processing capacity, for example it may be more efficient for a particular service to be carried out by a networked per sonal computer rather than a networked digital camera, where both of those devic es run the same or a similar service.

In an embodiment, the task manager 14 is configured to provide a task so that th e task can be stored in an incomplete form and transferred to another, for examp le more powerful, controller to enable completion of the next part of a task.

In an embodiment, the task manager 14 stores the current state of a task and the data associated in a context which can be passed between controllers allowing a user to access the task from different client devices. This means that a user can commence a task at one client device and then move to another that may be mo re suitable for continuation of the task with, in each case, the controller loca l to the user's current client device controlling the coordination and processin g of a task so as to enable the task data to be stored for later retrieval or pa ssed to another controller for continued processing.

In an embodiment, the services and tasks are defined using XML (Extensible Mark-up Language) and the task is stored in a context enabling parts of the task to b e passed between controllers.

Appendix 1 illustrates an example of a service description for a web gallery ser vice which creates a web gallery from a series of images. The ellipsis indicate omitted data. As can be seen from Appendix 1, the name of the service is "web gallery" and the service is of the type "synthesize". This service description defines a required input, a number of outputs and a number of parameters. As ca n be seen, the input has an ID "image" and must be of a type MIME in the JPEG fo rmat while an output of ID "HTML" must be of type MIME in the HTML format and th e output of ID "fullimage" and "thumbimage" must be of type MIME and in the JPEG format. The parameters in this example are title, thumbs per row, thumb width, thumb height, web file name, web images directory and web thumb prefix all of t ype "simple" with the title, web file name, web images directory and web thumb p refix being of "string" format and the remaining parameters being of "int" or in teger format.

Appendix 2 illustrates an example of a task description. In this example, the purpose of the task is to scan a document and send it via email and the task name is "Send OCR". As can be seen from Appendix 2, the task consists of address book, scanner, PDF conversion, email and OCR services with the respective service IDs "address", "service 2", "convert", "service 0" and "service 1". In addition, the task includes a user interface service of name "OCR email UI" that is to be downloaded to the client device when the services of ID "service 0", "service 1" and "convert" are being run.

When this task is implemented by a task manager, the task manager 14 first requests the operations controller 11 to cause the address book user interface service to be run to cause the address book user interface to be downloaded the client device to enable the user to select the name and address of the intended recipient of the email. Once the name and address of the intended recipient have been selected or entered by the user, then the task manager 14 requests the operation controller 11 to cause the scanner service to run. The scanner service enables a hard copy document placed on a scanner by the user to be scanned at a default resolution, in this example 300 DPI, and to provide output data with an ID "image" for input into the "convert" service. The scanner service does not require a user interface to run as the parameters required for the scanning, e.g. scan resolution are supplied by the task defaults. Once the task manager determines that the scanner service has provided the output data, then the task manager 14 requests the operations controller 11 to cause the user interface service "OCR Email UI" to be run and downloaded to the client device to enable input of the data required from the user for the remaining services. Once the task manager 14 has determined that the user has input the necessary information, in this case whether the user wants to send an OCRed version of the document to the email recipient, and/or send it in a PDF file which has been converted from the image, then the task manager 14 requests the operations controller 11 to run either the "convert" service, or the "OCR" service, or both. Once the task manager 14 determines that the "convert" and "OCR" services have completed their operations, if required, then the task manager 14 requests the operations controller 11 to cause the email service to run so as to enable the document to be emailed to the intended recipient.

As will be appreciated from Appendix 2 the operations controller 11 and task manager 14 coordinate carrying out of the task so that each service is called as and when required and provides output data for a target service with a given target service ID and target input ID.

The service discoverer 12 may discover services only upon instruction by the operations controller. As another possibility, the service discoverer 12 may be configured to check continually for new services on the system and to alert the operations controller 11 whenever new services are discovered. This would have the advantage that all services discoverable on the network would be immediately available to the task manager. However, where the network provides a large number of services, then this would require significant memory capacity to retain the descriptions of all of these services and may not be desirable where the controller has limited resources. Of course, one or more of the controllers on the ne

twork may be configured to discover services only upon request while one or more others of the controllers on the network may be configured continually to discover services, dependent upon the capabilities and requirements of the controllers.

In the described example, the user interfaces with the system via a single client device. As set out above, it may in some circumstances be desirable for the task to be portable within the network system. This may be achieved by storing the task in a data structure known as a context, for example an XML-based context which can be transferred to another controller or client device that may even use a different operating system and language without the requirement for translation of the context.

Thus, although not shown in Appendix 2, the task manager 14 may configure the task so as to enable the history of the task and the data associated with the current status of the task to be stored in an XML context so that once a service has run, the task may be stored with the data resulting from the running of that service so that it can, if required by the user, then be transferred to another controller or client device that can control the carrying out of the next service. In the embodiments described above, the service discoverer uses a plug-in architecture. Although this has advantages for the reasons set out above, the service discoverer 12 may also be implemented as a dedicated service discoverer capable of discovering only services operating with certain protocols. Similarly, in the embodiments described above, the task manager is implemented as a module which can be replaced or upgraded as required. Although this has advantages as set out above, it should also be possible to integrate the task manager within the operations controller, although this would make upgrading or replacing of the task manager more difficult.

APPENDIX 1

EXAMPLE SERVICE DESCRIPTION

```
<?xml version="1.0" encoding="UTF-8" ?>
- <CIAServiceGroup>
- <CIAService name="WebGalleryService" type="Synthesize"
    GUID="4f1d0d:fbee5bda43:-7ff2">

    <URI>class:WebGalleryService.jar:com.canon.cre.cia.services.webgall
    ery.WebGalleryService</URI>
- <Description>
  <Icon type="mime" format="image/png">iVBOR... QmCC</Icon>
  <Info lang="en" name="Web Gallery">This Service creates a web gallery
    from a series of images</Info>
  <Info lang="fr"
    name="Galerie Web">Ce service crée un galerie web d'une serie
  d'images</Info>
    </Description>
- <Inputs>
  <Input id="image" type="mime" format="image/jpeg" multiple="true"
    required="true" />
    </Inputs>
- <Outputs>
  <Output id="html" type="mime" format="text/html" multiple="true" />
  <Output id="fullImage" type="mime" format="image/jpeg" multiple="true"
    />
  <Output id="thumbImage" type="mime" format="image/jpeg"
    multiple="true" />
    </Outputs>
- <Parameters>
  <Parameter id="title" type="simple" format="string" />
  <Parameter id="thumbsPerRow" type="simple" format="int" />
  <Parameter id="thumbWidth" type="simple" format="int" />
  <Parameter id="thumbHeight" type="simple" format="int" />
  <Parameter id="webFileName" type="simple" format="string" />
  <Parameter id="webImagesDirectory" type="simple" format="string" />
  <Parameter id="webThumbPrefix" type="simple" format="string" />
    </Parameters>
    </CIAService>
- <CIAUIService name="WebGalleryUI" type="SynthesizeUI"
    GUID="4f1d0d:fbee5bda43:-7ff1">
  <Service name="WebGalleryService" />
  <UILanguage name="java" />
    </CIAUIService>
    </CIAServiceGroup>
```

APPENDIX 2

EXAMPLE TASK DESCRIPTION

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
UTF-8 Chars: □□□
  -->
- <CIATask name="SendOCR">
  <Device name="iR" />
  <Output uri="address" id="address" type="simple" format="string" />
- <Description>
  <Info lang="en" name="AddressBook" />
    </Description>
- <Description id="email">
  <Icon uri="scan.png" />
  <Info lang="en" name="Scanned Document">Scan a document and sends
    via email</Info>
   <Info lang="fr"
    name="Email le document">Scanner un document et envoyer par email</Info>
    </Description>
- <Description id="scan">
  <Icon uri="email.png" />
  <Info lang="en" name="Email Document">Scan a document and sends via
    email</Info>
   <Info lang="fr" name="Email le document">Scanner un document et envoyer
    par email</Info>

     </Description>
- <Service id="address" name="OpenAddressBook" type="ContactOutput">
  <Output id="address" required="true" multiple="true"
    targetServiceID="service0" targetInputID="address" />
    </Service>
- <Service id="service2" name="ScannerService" type="Scanner">
  <Parameter id="resolution" type="simple" format="int">300</Parameter>
  <Output id="image" required="true" multiple="true"
    targetServiceID="convert" targetInputID="inputData" />
  <Output id="image" required="true" multiple="true"
    targetServiceID="service1" targetInputID="image" />
    </Service>
- <Service id="convert" name="ConvertToPDFService" type="Convert">
  <Input id="inputData" required="true" multiple="false" />
  <Output id="outputData" required="true" multiple="true"
    targetServiceID="service0" targetInputID="attachments" />
    </Service>
- <Service id="service0" name="EmailService" type="Email">
  <Parameter id="subject" type="simple" format="string">Document from
    %fullname%</Parameter>
  <Input id="subject" required="false" multiple="false" />
```

```
<Input id="body" required="false" multiple="false" />
<Input id="address" required="true" multiple="true" />
<Input id="attachments" required="false" multiple="true" />
</Service>
- <Service id="service1" name="OcrService" type="Ocr">
<Input id="image" required="true" multiple="true" />
<Output id="text" required="true" multiple="true"
targetServiceID="service0" targetInputID="body" />
</Service>
- <UI name="OCREmailUI">
<Service id="service0" />
<Service id="service1" />
<Service id="convert" />
</UI>
- <UI name="AddressBookUI">
<Service id="address" />
</UI>
</CIATask>
```

5. Brief Description of the Drawings

Figure 1 shows a functional block diagram of a network system including a controller embodying the present invention for facilitating the carrying out of a task;

Figure 2 shows a more detailed functional block diagram of a controller embodying the present invention;

Figure 2a shows a very diagrammatic representation of a service description;

Figure 3 shows a functional block diagram of one example of computing apparatus configured to provide the controller shown in Figures 1 and 2;

Figure 4 shows a flowchart for illustrating one way in which a task manager of the controller shown in Figures 1 and 2 can manage the carrying out of a task;

Figure 5 shows a flowchart for illustrating operation of the controller to discover a new service;

Figure 6 shows a flowchart for illustrating operation of the controller to access a service;

Figure 7 shows a flowchart for illustrating operation of the controller in response to a request for a service; and

Figure 8 shows a flowchart illustrating another way in which the task manager can manage the carrying out of a task.

1. Apparatus connectable to a network for enabling the carrying out of task for a user, the apparatus comprising:

a service discoverer operable to discover services at locations on the network;

a task manager operable to control the carrying out of tasks using services discovered by the service discoverer;
a operations controller operable to control operations related to services; and
a service publisher operable, as a result of a discovery of a service by the service discoverer, to make that service discoverable by other apparatus connected to the network, wherein the service discoverer is arranged to discover services that use different protocols, and has a plug-in architecture.

2. Apparatus according to claim 1, wherein the operations controller is operable to receive requests for the running of services, to determine whether the requested service is a service local to the apparatus that the operations controller can run and, if not, to pass the request for the running of the service onto the location on the network at which the service discoverer discovered the service.

3. Apparatus according to claim 2, wherein the operations controller is configured to receive requests for the running of services from the task manager and from other apparatus connected to the network.

4. Apparatus according to claim 2 or 3, wherein the operations controller is operable to determine whether or not a request for a service to be run can be executed or passed onto to the location on the network at which the service discoverer discovered the service in accordance with at least one security rating associated with the at least one of the user and the requesting apparatus or task manager.

5. Apparatus according to claim 4, wherein the operations controller is configured to use different security ratings for requests from the task manager and requests from other apparatus for a service to be run.

6. Apparatus according to claim 4, wherein the operations controller is configured to implement different security restrictions depending upon whether the user is local to the apparatus or whether a service is being requested by another apparatus.

7. Apparatus according to any of the preceding claims, wherein the service publisher is configured to store a list of service descriptions each defining a function or functions implemented by the corresponding service.

8. Apparatus according to any of claims 1 to 6, wherein the operations controller is configured to receive and store for services discovered by the service discoverer service descriptions with each service description defining a function or functions implemented by the corresponding service.

9. Apparatus according to claim 7 or 8, wherein the service publisher is configured to store a list of the service descriptions discovered by the service discoverer.

10. Apparatus according to any of claims 7 to 9, wherein each stored service description includes at least a service name, a description of the function of the service, and input, output and parameter data required by the service.

11. Apparatus according to any preceding claim, wherein the task manager is operable to select a service from a number of different discovered services that can perform the same function on the basis of predetermined criteria.

12. Apparatus according to any of claims 1 to 10, wherein the task manager is operable to select a service from a number of different discovered services that can perform the same function on the basis of at least one of the following criteria: an indication of the location on the network of the service; a rating system; a required processing time; a current status of the services; a service running cost; a service quality; and a user recommendation.

13. Apparatus according to any preceding claim, wherein, in the event that the same service is discovered more than once by the service discoverer by different routes over the network, the operations controller is operable to store information related to each of the routes to the service to enable the operations controller to access the service by a different route.

14. Apparatus according to claim 13, wherein the operations controller is operable to select another route to a service in the event that the same service is discovered more than once by the service discoverer by different routes over the network and a request via one of the routes to run the service fails.

15. Apparatus according to any preceding claim, wherein, upon discovery of a service, the apparatus is operable to increment a hop count associated with that service and indicating the number of steps from one apparatus to another over the network to the location of the service.

16. Apparatus according to any of claims 7 to 10, wherein, upon discovery of a service, the apparatus is operable to increment a hop count indicating the number of steps from one apparatus to another over the network to the location of the service, wherein the service descriptions are associated with route information that indicates the route taken over the network from the service location to the apparatus and wherein the event that the service discoverer discovers a service more than once, the operations controller is operable to select a route on the basis of the associated hop count.

17. Apparatus according to any preceding claim, wherein the service discoverer comprises service discovery modules and a service discovery manager operable to manage said modules.

18. Apparatus according to claim 17, wherein one service discovery module is configured to discover Web services, another service discovery module is configured to discover JINI services and another service discovery module is configured to discover local services.

19. Apparatus according to any of the preceding claims, wherein the task manager is configured to provide a task so that the task can be stored in an incomplete form and transferred to another apparatus.

20. Apparatus according to claim 19, wherein the task manager is configured to s tore a current state of a task and data associated with the task in a context, f or example an XML context, which can be passed between apparatus.

21. Apparatus according to any of the preceding claims, wherein the operation co ntroller 11 is operable to filter out services that cannot be used by a client d evice or client devices coupled to the apparatus.

22. Apparatus according to any preceding claim in combination with at least one client device via which a user can interface with the apparatus.

23. Apparatus according to any preceding claim in combination with storage means storing at least one service.

24. A network system apparatus comprising a plurality of apparatus in accordance with any of claims 1 to 20, at least one client device associated with each app aratus, and a plurality of service storage means storing services, at least some of the apparatus being directly connectable to the network and at least one app aratus being connectable to the network via another apparatus and at least some of the service storage means being connected to the network only via an apparatu s in accordance with any of claims 1 to 20.

25. Apparatus according to any preceding claim, wherein the services have a comm on application programming interface.

26. Apparatus according to claim 23, 24 or 25, wherein the services include inpu t services, output services, and processing services.

27. Apparatus according to claim 23, 24 or 25, wherein the services include inpu t services, output services, processing services and user interface services.

28. Apparatus according to any of claims 23 to 27, wherein at least some service s have subsidiary services.

29. Apparatus according to any of claims 23 to 28, wherein at least some service s have default values for parameters and the task manager is configured to enabl e a default value to be provided for a task that overrides a service default val ue.

30. Apparatus according to claim 27, wherein at least one of some services and s ome tasks have default values for at least one parameter and at least one user i nterface service is configured to enable a user to specify a value for a paramet er that overrides the default value.

31. A method of enabling the carrying out of task for a user, the method compris ing apparatus carrying out the steps of:
discovering services at locations on the network by using a plug-in architectur e to discover services that use different protocols;
controlling the carrying out of tasks using discovered services; and
as a result of the said discovery making discovered services discoverable by ot

her apparatus connected to the network.

32. A method according to claim 31, further comprising the apparatus carrying ou t the steps of receiving requests for the running of services, determining wheth er the requested service is a service local to the apparatus that can be run and , if not, passing the request for the running of the service onto the location o n the network at which the service was discovered.

33. A method according to claim 31 wherein requests for the running of services are received from a task manager of the apparatus and from other apparatus conne cted to the network.

34. A method according to claim 32 or 33, further comprising the apparatus carry ing out the steps of determining, in accordance with at least one security ratin g associated with the at least one of the user and the requesting apparatus or t ask manager, whether or not a request for a service to be run can be executed or passed onto to the location on the network at which the service was discovered.

35. A method according to claim 34, wherein different security ratings are used for requests from a task manager of the apparatus and requests from other appara tus for a service to be run.

36. A method according to claim 34, wherein different security restrictions are implemented depending upon whether the user is local to the apparatus or whether a service is being requested by another apparatus.

37. A method according to any of claims 31 to 36, further comprising the apparat us storing a list of service descriptions each defining a function or functions implemented by the corresponding service.

38. A method according to any of claims 31 to 36, further comprising the apparat us receiving and storing for discovered services service descriptions with each service description defining a function or functions implemented by the correspo nding service.

39. A method according to claims 37 or 38, wherein each stored service descripti on includes at least a service name, a description of the function of the servic e, and input, output and parameter data required by the service.

40. A method according to any of claims 31 to 39, wherein the apparatus selects a service from a number of different discovered services that can perform the sa me function on the basis of predetermined criteria.

41. A method according to any of claims 31 to 39, wherein the apparatus selects a service from a number of different discovered services that can perform the sa me function on the basis of at least one of the following criteria: an indicatio n of the location on the network of the service; a rating system; a required pro cessing time; a current status of the services; a service running cost; a servic e quality; and a user recommendation.

42. A method according to any of claims 31 to 39, wherein, in the event that the same service is discovered more than once by different routes over the network, the apparatus stores information related to each of the routes to the service to enable the service to be accessed by a different route.

43. A method according to claim 42, wherein the apparatus selects another route to a service in the event that the same service is discovered more than once by different routes over the network and a request via one of the routes to run the service fails.

44. A method according to any of claims 31 to 43, wherein, upon discovery of a service, the apparatus increments a hop count indicating the number of steps from one apparatus to another over the network to the location of the service.

45. A method according to any of claims 37 to 39, wherein, upon discovery of a service, the apparatus increments a hop count indicating the number of steps from one apparatus to another over the network to the location of the service, wherein the service descriptions are associated with route information that indicates the route taken over the network from the service location to the apparatus and wherein, in the event that a service is discovered more than once, the apparatus selects a route on the basis of the associated hop count.

46. A method according to any of claims 31 to 45, wherein the apparatus comprises service discovery modules and the method further comprises managing said modules.

47. A method according to claim 46, wherein one service discovery module discovers Web services, another service discovery module discovers JINI services and another service discovery module discovers local services.

48. A method according to any of claims 31 to 47, wherein the apparatus configures a task so that the task can be stored in an incomplete form and transferred to another apparatus.

49. A method according to claim 48, wherein the apparatus stores a current state of a task and data associated with the task in a context, for example an XML context, which can be passed between apparatus.

50. A method according to any of claims 31 to 49, wherein the services have a common application programming interface (API).

51. A method according to any of claims 31 to 50, wherein the services include input services, output services, and processing services.

52. A method according to any of claims 31 to 51, wherein the services include input services, output services, processing services and user interface services.

53. A method according to any of claims 31 to 52, wherein at least some services have subsidiary services.

54. A method according to any of claims 31 to 53 wherein at least some services have default values for parameters and the apparatus enables a default value to be provided for a task that overrides a service default value.

55. A method according to claim 52, wherein at least one of some services and so me tasks have default values for at least one parameter and at least one user in terface service enables a user to specify a value for a parameter that overrides the default value.

56. A method according to any of claims 31 to 55, further comprising the apparat us filtering out services that cannot be used by a client device or client devic es coupled to the controller.

57. A computer program product comprising program instructions for programming a processor to carry out a method in accordance with any of claims 31 to 56.

58. A storage medium storing program instructions for programming a processor to carry out a method in accordance with any of claims 31 to 56.

59. Apparatus connectable to a network for enabling the carrying out of task for a user, the apparatus comprising:
a service discoverer operable to discover services at locations on the network;
a task manager operable to control the carrying out of tasks using services disc overed by the service discoverer;
an operations controller operable to control operations related to services; and
a service publisher operable, as a result of a discovery of a service by the ser vice discoverer, to make that service discoverable by other apparatus connected to the network, wherein the service discoverer comprises a number of service dis covery plug-in modules and a service discovery manager for managing the modules, with the different service discovery plug-in modules being arranged to discover services that use different protocols.

1. Abstract

In a network system, a client device is connected to the network via a controlle r that enables a user of the client device to carry out a task using one or more services available on the network.The controller (10) has a service discoverer (12) that discovers services at locations on the network (N) and a task manager (14) that controls the carrying out of tasks using services discovered by the se rvice discoverer (12). The controller (10) also has a service publisher (13) th at, as a result of the discovery of a service by the service discoverer (12) mak es that service discoverable by other controllers (10) connected to the network, thereby allowing those other controllers to discover and access that service or services via that controller.

FIG.1

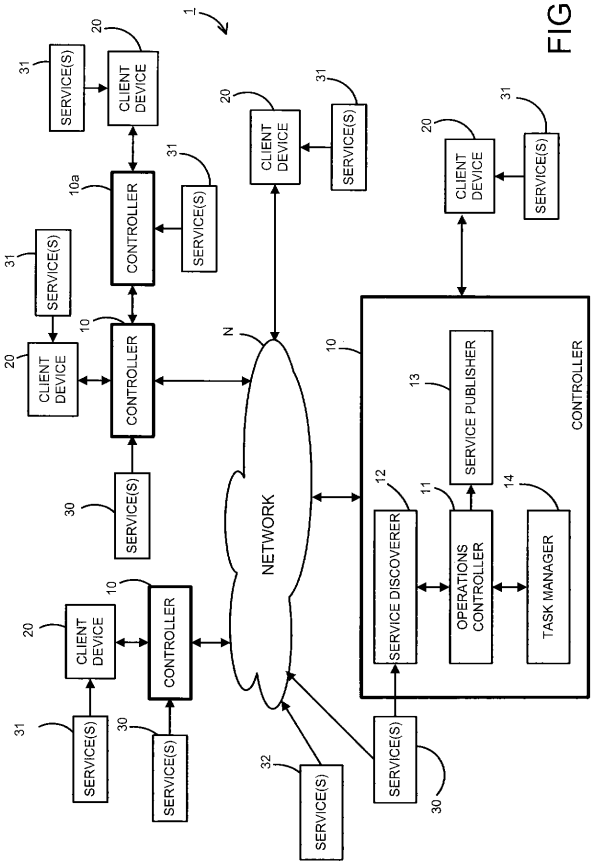
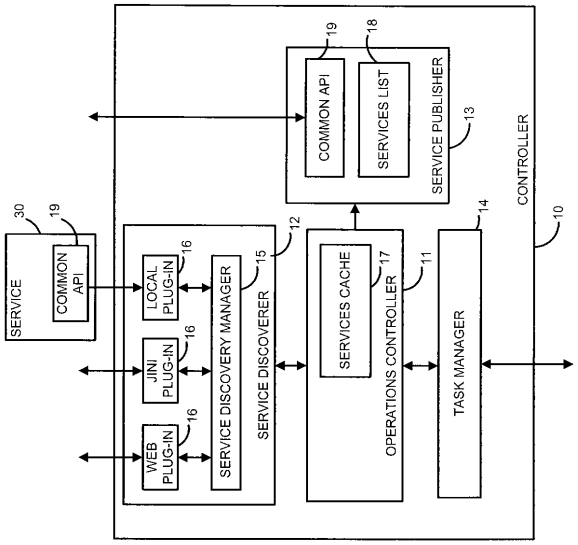


FIG.2

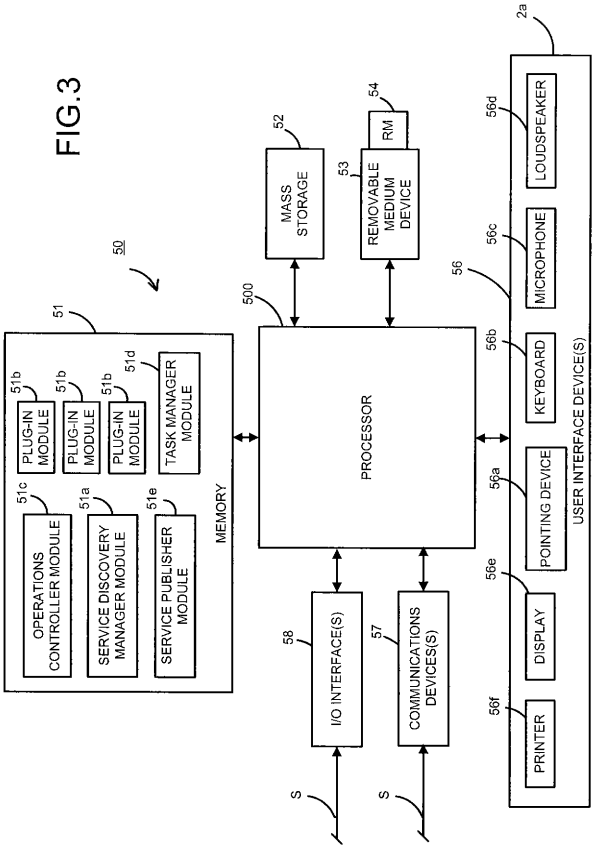


100

SERVICE	ROUTE INFORMATION
SERVICE NAME	
ID	
TYPE	
CLASSIFICATION	
INPUTS	
OUTPUTS	
PARAMETERS	
CAPABILITIES	

FIG.2a

FIG.3



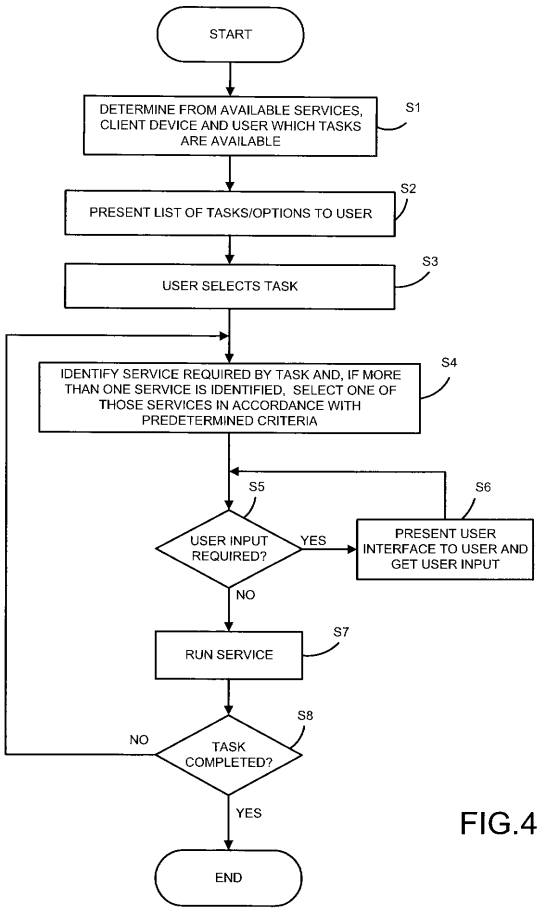


FIG.4

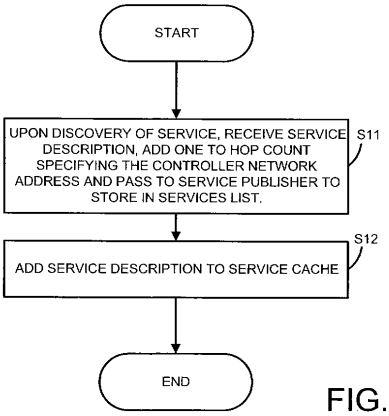


FIG.5

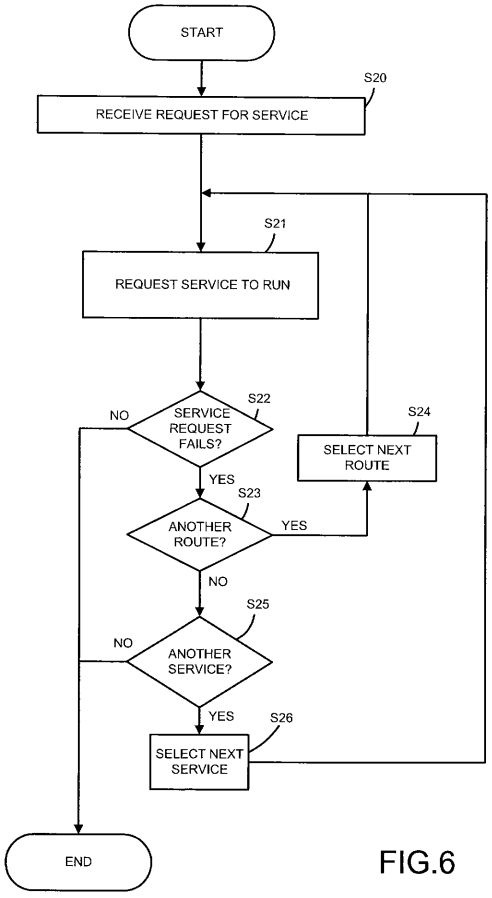


FIG.6

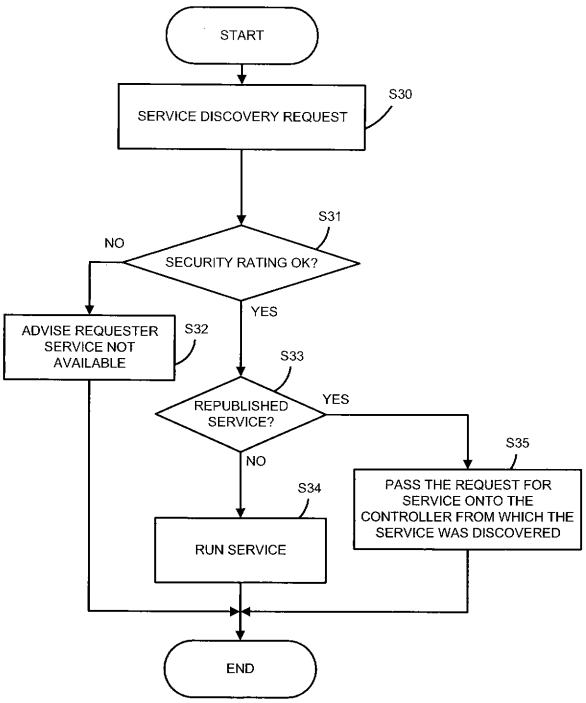


FIG.7

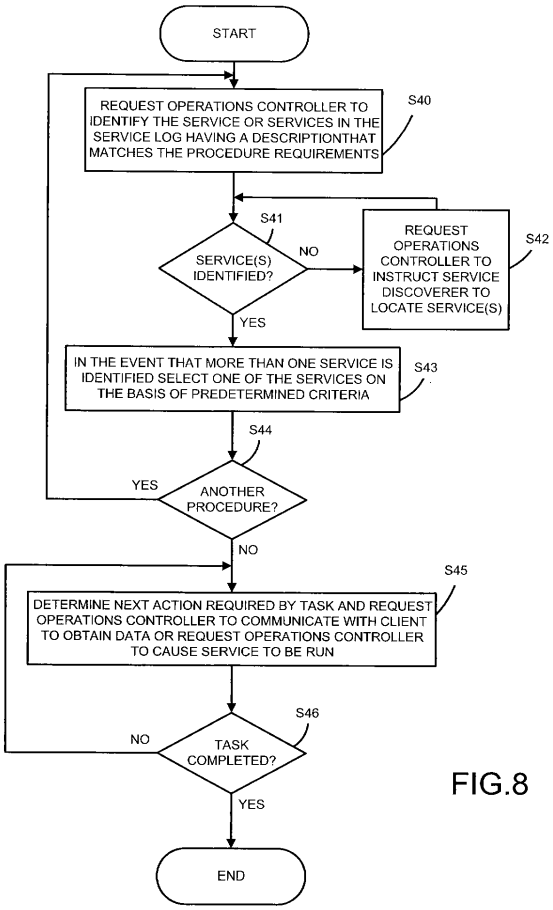

START
REQUEST OPERATIONS CONTROLLER TO IDENTIFY THE SERVICE OR SERVICES IN THE SERVICE LOG HAVING A DESCRIPTIONTHAT MATCHES THE PROCEDURE REQUIREMENTS
S40
S41
SERVICE(S) IDENTIFIED?
NO
REQUEST OPERATIONS CONTROLLER TO INSTRUCT SERVICE DISCOVERER TO LOCATE SERVICE(S)
S42
YES
IN THE EVENT THAT MORE THAN ONE SERVICE IS IDENTIFIED SELECT ONE OF THE SERVICES ON THE BASIS OF PREDETERMINED CRITERIA
S43
S44
YES
ANOTHER PROCEDURE?
NO
S45
DETERMINE NEXT ACTION REQUIRED BY TASK AND REQUEST OPERATIONS CONTROLLER TO COMMUNICATE WITH CLIENT TO OBTAIN DATA OR REQUEST OPERATIONS CONTROLLER TO CAUSE SERVICE TO BE RUN
S46
NO
TASK COMPLETED?
YES
END

FIG.8