

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2003-208402
(P2003-208402A)

(43) 公開日 平成15年7月25日 (2003.7.25)

(51) Int.Cl. ⁷	識別記号	F I	ターマコード* (参考)
G 0 6 F 15/00	3 1 0	G 0 6 F 15/00	3 1 0 R 5 B 0 8 5

審査請求 未請求 請求項の数 3 OL (全 13 頁)

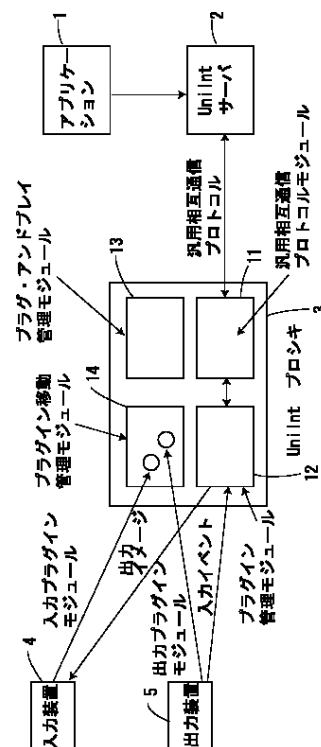
(21) 出願番号 特願2002-8364(P2002-8364)	(71) 出願人 899000068 学校法人早稲田大学 東京都新宿区戸塚町1丁目104番地
(22) 出願日 平成14年1月17日(2002.1.17)	(72) 発明者 中島 達夫 東京都新宿区大久保3-4-1 早稲田大 学理工学部内 (74) 代理人 100080089 弁理士 牛木 護 (外2名) Fターム(参考) 5B085 BA06 BE07 CE06

(54) 【発明の名称】 ネットワーク機器汎用相互通信装置

(57) 【要約】

【課題】 従来の標準グラフィカル・ユーザー・インターフェイスをそのまま利用して、一様な手順で機器の制御を行なう。

【解決手段】 利用可能な機器に応じたグラフィカル・ユーザー・インターフェイスがUniIntプロキシ3に送られると、出力装置5の表示手段に適合するように変換され、表示手段に転送される。また、利用者が利用可能な機器を制御するには、現在利用できる入力装置4から適宜コマンドを伝送すればよい。こうすれば、どのような入力装置4であっても、UniIntプロキシ3は伝送されたコマンドを、汎用相互通信プロトコルに適合するイベントに変換し、それをアプリケーション1に伝送する。



【特許請求の範囲】

【請求項1】 利用できる機器に応じて、その機器を制御するのに必要なグラフィカル・ユーザー・インターフェースを生成して提示するアプリケーション手段と、前記機器を制御するためのコマンドを伝送する入力装置と、表示手段を備えた出力装置と、前記入力装置および出力装置と前記アプリケーション手段との間に介在するプロキシ手段とを備え、前記プロキシ手段は、前記入力装置や前記出力装置から受信メッセージを受け取ることにより、現在利用可能な入力装置や出力装置を検出してそれを選択する利用可能装置検出手段と、前記アプリケーション手段から汎用相互通信プロトコルによって送られてきたグラフィカル・ユーザー・インターフェースに含まれるイメージを、前記利用可能装置検出手段が選択した前記出力装置の表示手段に表示できるように変換し、この変換したイメージを含むグラフィカル・ユーザー・インターフェースを該出力装置の表示手段に転送する出力管理手段と、前記利用可能装置検出手段が選択した前記入力装置からコマンドを受取ると、このコマンドを前記アプリケーション手段が認識できるイベントに変換し、この変換したイベントを汎用通信プロトコルによって該アプリケーション手段へ伝送する入力管理手段と、を備えたことを特徴とするネットワーク機器汎用相互通信装置。

【請求項2】 前記出力管理手段は、前記入力装置から所望の前記出力装置を選択するコマンドを受取ると、前記利用可能装置検出手段による選択に優先して、この出力装置の表示手段が表示できるグラフィカル・ユーザー・インターフェースを、該出力装置に転送するものであることを特徴とする請求項1記載のネットワーク機器汎用相互通信装置。

【請求項3】 前記アプリケーション手段は、現在利用可能な前記機器を特定するために、IEEE1394ネットワークを通じて該機器を接続したことを特徴とする請求項1または2記載のネットワーク機器汎用相互通信装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワーク化された家庭用機器や個人用機器との汎用的な相互通信を行うためのネットワーク機器汎用相互通信装置に関する。

【0002】

【発明が解決しようとする課題】我々の毎日の生活は、コンピュータを内蔵した種々の対象物により劇的に変化をするであろう。これらの対象物は、人々の身体と記憶を拡大するため知的な動作をするものと期待される。

【0003】こうしたコンピュータ環境を実現する為に、これまで多くの研究プロジェクトが種々の問題に挑

戦している。これらのコンピュータ環境は、広くユビキタス(Ubiquitous:どこにでもある)・コンピューティングと呼ばれている。また、他の研究者たちは、パーベシブ・コンピューティング("pervasive computing") [3]、センチエント・コンピューティング("sentient computing") 或いはシングス・ザット・シンク("Things That Think")と呼ばれる同様の概念を提案している。ユビキタス・コンピューティング環境においては、種々の対象物がコンピュータを内包することにより増加される。如何なるプログラムも、コンピュータによって実行することができるので、プログラムを置き換えることにより、これらの対象物は無限に拡張される可能性を持っている。また、これらの対象物は、他の対象物と通信を行なうネットワークを持つので、周囲を取り巻く状況に従って、各対象物がそれぞれでプログラムを置き換えることにより、一層アクティブに動作するであろう。例えば、我々の環境は我々に代わって世界中に起こっていることを記憶するかもしれないし、各々の対象物が現時点で存在する場所を我々に教えるかもしれない。

【0004】前述したように、前記ユビキタス・コンピューティング環境を実現する為に、多くの研究が成されている。或る研究グループは、ユビキタス・コンピューティング環境のプロトタイプ(代表例)を構築する作業を行ってきた。これらの研究は、ユビキタス・コンピューティングが我々の生活に与えるインパクトを示している。また最近は、幾つかの標準ミドルウェアの仕様が、ユビキタス・コンピューティングを実現するために提案されている。ミドルウェアとは、いわゆるOS(Operating System)上で動作し、アプリケーションソフトに対してOSよりも高度で具体的な機能を提供するソフトウェアのことで、OSとアプリケーションソフトの間的な性格を持っている。そのなかで、ネットワークを通じて各種機器を接続し、相互に機能を提供しあうための技術仕様であるJini(登録商標)や、同じように家庭内のパソコンや周辺機器、AV機器、電話、家電製品などの機器をネットワークを通じて接続し、相互に機能を提供しあうための技術仕様であるUPnP(ユニヴァーサル・プラグアンド・プレイ)は、ミドルウェアとして良く知られているが、これらは分散された環境の中で種々のサービスを発見する機構を備えている。また、別なミドルウェアとして知られるHAVI(Home Audio/Video Interoperability)は、ネットワーク化された家庭用機器の発展を可能にする。それ故に、家庭環境におけるユビキタス・コンピューティングは、近い将来に実現されるであろう。こうしたユビキタス・コンピューティング環境は、ホーム・コンピューティングと呼ばれるが、ホーム・コンピューティングは、我々の日常生活に直接関係があるので、とりわけ興味深い分野ではある。また、総ての人は、ホーム・コンピューティング環境の急激な進歩を期待しているので、ユビキタス・コンピューティングの効

果を示すことは的を得ていることである。

【0005】こうしたホーム・コンピューティング環境で最も重要な問題の一つは、コンピュータを内蔵した種々の対象物が、如何に相互通信し合うかということである。利用者と種々の対象物に内蔵されたコンピュータ間の相互通信は、従来から幾つかの研究グループ[2,13,17,30]により開発が行なわれていた。これらの装置によれば、より自然に内蔵したコンピュータと相互通信を行なうことができる。しかしながら、ネットワーク化された家庭用機器向けの現在の標準ミドルウェアは、Java（登録商標）に標準で付属するグラフィック関連のクラスライブラリとして知られるJava AWTや、同じようにグラフィック関係のライブラリとして知られるGTK+の様な、従来からある標準グラフィカル・ユーザー・インターフェースを採用してきた。それ故に、PDA、携帯電話あるいは上記研究段階のプロトタイプ相互通信機器から、家庭用の機器を制御することは容易なことではない。また、利用者の現在の状況によって、自然な相互通信は変化する。例えば、利用者が食事を料理しているならば、音声によって機器を制御したいが、ソファの上でTVを見ている場合は、リモート・コントロールを使うのが良いのかもしれない。このことは、最も適当な相互通信は、利用者の現在の状況と好みによってダイナミックに選定すべきであり、この相互通信装置の選定は、利用者が家庭内、事務所或いは公共の場所のように如何なる場所に居ようと、一貫したものであることを意味している。

【0006】そこで本発明は、利用者がいかなる場所にいたとしても、従来の標準グラフィカル・ユーザー・インターフェースをそのまま利用して、一様な手順で機器の制御を行なうことができるネットワーク機器汎用相互通信装置を提供することをその目的とする。

【0007】

【課題を解決するための手段】本発明の請求項1におけるネットワーク機器汎用相互通信装置は、利用できる機器に応じて、その機器を制御するのに必要なグラフィカル・ユーザー・インターフェースを生成して提示するアプリケーション手段と、前記機器を制御するためのコマンドを伝送する入力装置と、表示手段を備えた出力装置と、前記入力装置および出力装置と前記アプリケーション手段との間に介在するプロキシ手段とを備え、前記プロキシ手段は、前記入力装置や前記出力装置から受信メッセージを受け取ることにより、現在利用可能な入力装置や出力装置を検出してそれを選択する利用可能装置検出手段と、前記アプリケーションから汎用相互通信プロトコルによって送られてきたグラフィカル・ユーザー・インターフェースに含まれるイメージを、前記利用可能装置検出手段が選択した前記出力装置の表示手段に表示できるように変換し、この変換したイメージを含むグラフィカル・ユーザー・インターフェースを該出力装置の

表示手段に転送する出力管理手段と、前記利用可能装置検出手段が選択した前記入力装置からコマンドを受取ると、このコマンドを前記アプリケーション手段が認識できるイベントに変換し、この変換したイベントを汎用通信プロトコルによって該アプリケーション手段へ伝送する入力管理手段とを備えて構成される。

【0008】この場合、利用可能な機器に応じたグラフィカル・ユーザー・インターフェースが、アプリケーション手段から汎用相互通信プロトコルによってプロキシ手段に送られると、そのグラフィカル・ユーザー・インターフェースに含まれるイメージは、プロキシ手段により現在利用可能な出力装置の表示手段に適合するように変換され、その変換されたグラフィカル・ユーザー・インターフェースが表示手段に転送される。また、利用者が利用可能な機器を制御するには、表示手段に表示されるグラフィカル・ユーザー・インターフェースを参照しながら、現在利用できる入力装置から適宜コマンドを伝送すればよい。こうすれば、どのような入力装置であっても、プロキシ手段は伝送されたコマンドを、汎用相互通信プロトコルに適合する例えばマウスまたはキーボードのイベントに変換し、それをアプリケーション手段に伝送する。

【0009】つまり、ここでのプロキシ手段は異種の異種の入力装置や出力装置を同じように取り扱う手段としての機能を果すので、現存する標準のグラフィカル・ユーザー・インターフェースを変更することなく、種々の入力装置や出力装置を使って、一様な手順で機器の制御を行なうことができる。

【0010】また、本発明の請求項2におけるネットワーク機器汎用相互通信装置は、前記入力装置から所望の前記出力装置を選択するコマンドを受取ると、前記利用可能装置検出手段による選択に優先して、この出力装置の表示手段が表示できるグラフィカル・ユーザー・インターフェースを、該出力装置に転送するように、前記出力管理手段を構成したものである。

【0011】このようにすると、入力装置から表示したい出力装置を選択するコマンドを伝送すると、その出力装置の表示手段に適合したグラフィカル・ユーザー・インターフェースがプロキシ手段から送り出される。したがって、利用者の状況や好みに応じた出力装置の切替えが可能となり、利用者と機器との間を独自のものとすることができる。

【0012】本発明の請求項3におけるネットワーク機器汎用相互通信装置は、前記アプリケーション手段が現在利用可能な前記機器を特定するために、IEEE1394ネットワークを通じて該機器を接続したことを特徴とする。

【0013】IEEE1394ネットワークは、どの機器が現在接続されているか、誰のパワースイッチがオンになっているかを知らせる機構をサポートしているので、アプリケーション手段が現在利用可能な機器を容易に知ること

ができると共に、最も相応しいユーザー・インターフェイスを容易に選定することができる。

【0014】

【発明の実施形態】以下、添付図面を参照しながら、本発明における好適な実施例を説明する。先ず実施例の詳細を説明する前に、ホーム・コンピューティング環境において、利用者との相互通信すなわち対話をサポート（支援）する幾つかの設計選択について説明する。

【0015】本発明の目的を達成するための方法は数種ある。最も重要な論点は、種々の相互通信装置（デバイス）を如何に支援するかである。従来のユーザー・インターフェイス・システムは、数種の相互通信装置のタイプを想定している。例えば、マウス、トラック・ポイントそしてタッチ・スクリーンは、入力装置として使用される。柔軟なユーザー・インターフェイス・システムを構築する手法としては、次の3つの代案が考えられる。

(1) 零の状態から新規のユーザー・インターフェイス・システムを構築する。

(2) 従来からある種々の相互通信装置用の各ユーザー・インターフェイス・システムを呼び出す新たな装置の独立層（レイヤ）を構築する。

(3) 従来のユーザー・インターフェイス・システムの入力/出力イベントを獲得し、相互通信装置に従ってそのイベントを翻訳する。

【0016】(1)の手法は、最初から種々の相互通信装置によって使用される新規なユーザー・インターフェイス・システムの構築を意味する。この手法は、システムが新たな相互通信装置に適応されることを考慮に入れるので、新相互通信装置が出現した時に生き長らえることができる。しかしながら、現存の機器とミドルウェアのコンポーネント（構成要素）を新たなユーザー・インターフェイス・システムに編入する為には修正が必要があり、その開発には長い期間を必要とする。多くのホーム・コンピューティング・システムは、すでに現存のソリューション（問題点の解決や要求の実現を行なうための情報システム）を採用しているため、現在使用されているユーザー・インターフェイス・システムを置き換えることは不可能である。

【0017】(2)の手法は、従来のユーザー・インターフェイス・システム上に新しい層を追加することにより、種々の相互通信装置を支援する2つの案がある。第1案は、相互通信装置の独立層が、アプリケーションの標準的な要求を各ユーザー・インターフェイス・システムへの要求に翻訳する。例えば、こうした手法に基礎を置いた文献の中では、利用者との間の相互通信方法を含むドキュメントを、アプリケーションが指定している。装置の独立層は、現存する各ユーザー・インターフェイス・システム用のドキュメントを提供する。第2案であるマルチプル・ユーザー・インターフェイス・システムにおいて、各機器は、getUI()法を提供する。こ

の方法は、マルチプル・ユーザー・インターフェイス・サポート・サービスへ参照信号を返し、各相互通信装置に最も適したユーザー・インターフェイスを提供する。双方の案は、非常に将来性があるが、第1案の様に現存のアプリケーションを修正する必要があり、装置の独立層がそれぞれの相互通信装置に対し実行されなければならない。それ故に、この手法は、ホーム・コンピューティング用に現存する標準仕様を支援するには適当ではない。

【0018】最後の(3)の手法は、ユーザー・インターフェイス・システムによって生成されたビットマップ・イメージが、機器制御のために相互通信装置の特性に従って変換される。また、相互通信装置からの入力イベントは、マウス或いはキーボードのイベントに変換される。この手法では、入力/出力イベントの変換のために、グラフィカル・ユーザー・インターフェイス（GUI）内のレイアウト情報を利用できないという制限がある。しかし、大部分の家庭用音響映像機器は、グラフィカル・ユーザー・インターフェイスを提示する表示装置を備えている。この場合の最も重要な要求とは、グラフィカル・ユーザー・インターフェイスを提示する表示装置と、利用者の状況や好みに応じて、グラフィカル・ユーザー・インターフェイスを提示する入力装置を変えたいということであるが、こうした要求は、(3)の手法において非常に上手く達成できる。一方、この手法は、ホーム・コンピューティング用の現存のソフトウェアを修正する必要は無い。また、従来のホーム・コンピューティング用標準ミドルウェアで採用されてきたJava AWT/Swing, X window system, Microsoft Windows（登録商標）のような人気のあるユーザー・インターフェイス・システムの大部分は、本手法にて極めて容易に採用できる。

【0019】ここでは、HAViを実行するホーム・コンピューティング・システムがJava AWTの使用を必要としており、しかも近い将来に、修正を行なうことなくHAViに基づき開発される種々のホーム・コンピューティング・アプリケーションを使いたいという前提で、(3)の手法に基づく実施例を説明する。特に最近の興味は、家庭用機器を構築するためのLinuxを内蔵した使用形態にあって、それは(3)の手法をより実際的なものにする。何故なら、Linuxはその基本的ユーザー・インターフェイス・システムとして、X windowsシステムを通常は取り入れており、内蔵したLinux上の種々のアプリケーションが、種々の相互通信装置により制御されることを、本実施例で提案するシステムが可能にするからである。こうしたアプリケーションは、従来の良く知られているユーザー・インターフェイス・ツールキットを使用可能であるため、この手法は、種々の内蔵アプリケーションを極めて容易に発展させることができる。

【0020】本実施例で以後説明する(3)の手法にお

いて、入力/出力相互通信装置と機器との間の通信に広く利用できるプロトコルを、ここで汎用相互通信 ("universal interaction") と呼ぶ。この汎用相互通信は、均一な方法にて種々の家庭用機器を制御することができる。このことは、何処にいて何を制御したいかによって、人々の行動が制限を受けないことを意味する。したがって、ここでの手法は、家庭用機器との極めて自然な相互通信を提供する。

【0021】機器からの出力イベントは汎用出力相互通信イベントに変換され、そのイベントは、それぞれの出力相互通信装置に向けて翻訳される。また、入力相互通信装置で発生する入力イベントは汎用入力相互通信イベントに翻訳され、そのイベントは機器内で実行されるアプリケーションにより処理される。

【0022】後述するUniIntプロキシと呼ばれる汎用相互通信プロキシは、汎用相互通信プロトコルと各相互通信装置の入力/出力イベント間の変換を広範囲に行なう役割を演じる。もしも、各入力/出力相互通信装置のイベントが汎用相互通信プロトコルに変換されるならば、プロキシは機器の制御のために如何なる入力/出力相互通信装置の使用も可能にする。この手法は、次のような三つの非常に魅力的な特徴を提供する。

【0023】第1の特徴は、入力相互通信装置と出力相互通信装置が、利用者の状況と好みによって、それぞれ独自に選択されることである。例えば、利用者は、PDAを入力/出力相互通信装置に選択できる。また、利用者は、携帯電話を入力相互通信装置に選択してもよいし、またテレビジョンも出力相互通信装置として選択してもよい。利用者は、携帯装置により生じる拡大された実世界を操作することで、自分の意思表示によって機器を制御できる。

【0024】第2の特徴は、利用者の好みに応じてそれに相応しい入力/出力相互通信装置を選択できることである。しかも、これらの相互通信装置は、利用者の現在の状況によってダイナミックに変更される。例えば、入力相互通信装置として携帯電話を使用している利用者は、別の仕事のために現在両手が塞がっているので、相互通信装置を音声入力システムに変えるであろう。

【0025】第3の特徴は、もしもユーザー・インターフェース・システムが汎用相互通信プロトコルと交信できれば、機器内で実行中の如何なるアプリケーションでも、あらゆるユーザー・インターフェース・システムを利用できることである。例えば、キーボード/マウスを汎用入力イベントとして採用し、ビットマップ・イメージを汎用出力イベントとして採用した場合、あらゆる相互通信装置との接続のために、Java AWT, GTK+ そしてQtのような従来のグラフィカル・ユーザー・インターフェースのツールキットを利用することができる。

【0026】実際、家庭用電化製品の殆どの標準仕様は、GUI(グラフィカル・ユーザー・インターフェース)

標準用のJava AWTを最近では採用する傾向にある。かくして、本手法を取り入れたシステムは、将来の各種家庭用電化製品を、種々の相互通信装置から、それらのアプリケーションを修正することなく制御することが可能になるであろう。現存のGUI標準を変更することは非常に難しいので、この特徴は非常に望ましいものである。

【0027】本実施例におけるシステムは、グラフィカル・ユーザー・インターフェースを描くためにビットマップ・イメージを転送すると共に、入力用マウス/キーボードのイベントを処理するために、シンクライアント・システム ("thin-client system") を使用する。通常のシンクライアント・システムは、ビューアとサーバから成る。サーバは、アプリケーションが動作する機器上で実行される。アプリケーションは、X window system の様な従来のユーザー・インターフェースを使って、グラフィカル・ユーザー・インターフェースを実行する。ユーザー・インターフェース・システムにより作られるビットマップ・イメージは、通常はもう一つの機器上で実行されるビューアへ送信される。一方、ビューアによって読み込まれるマウスとキーボード・イベントは、サーバへ転送される。ビューアとサーバ間のプロトコルは、標準プロトコル(例えば、前記VNCシステムに於いては、RFB (Remote Frame Buffer) プロトコルと呼ばれる)として規定される。ここでは、こうしたプロトコルを汎用相互通信プロトコル (universal interaction protocol) と呼ぶ。本システムは、利用者の居場所によって利用者の表示部(デスクトップ)を動かしたり、例えばMS Windows(登録商標)とX Window systemの両方とるように、同一画面上に複数の表示部を表示する。

【0028】本システムでは、シンクライアント・システムのビューアを、UniInt (Universal Interaction: 汎用相互通信) サーバから受け取ったビットマップ・イメージを出力装置へ転送するUniIntプロキシに置き換える。この手法では、如何なるシンクライアント・システムのサーバも、UniIntサーバとして使用可能である。また、UniIntプロキシは、入力相互通信装置から受け取った入力イベントをUniIntサーバへ転送する。

【0029】本実施例におけるシステムは、図1に示すように、次の主要な5つの構成要素から成る。すなわちそれは、家庭用機器(図示せず)のアプリケーション1と、前記UniIntサーバ2と、UniIntプロキシ3と、入力装置4と、出力装置5である。

【0030】表示操作制御手段に相当する家庭用機器のアプリケーション1は、現在使用できる家庭用機器を制御するのに必要なグラフィカル・ユーザー・インターフェース(表示操作形態)を作成するものである。例えば、家庭用機器としてTV(テレビジョン)が現在利用できるならば、アプリケーション1は、TV用のユーザー・インターフェースを作成する。一方、アプリケーション1は、いずれも家庭用機器であるTVとVCR(デジタル

ビデオカメラ)が現在両方とも利用できるならば、TVとVCR用のGUIを構成する。

【0031】汎用相互通信サーバに相当するUniIntサーバ2は、windowシステムによって作られるビットマップ・イメージを、汎用相互通信プロトコルを使って、UniIntプロキシ3へ送り出すと共に、UniIntプロキシ3から受け取ったマウスとキーボードの各イベントを、windowシステムに転送する機能を有する。現在の実行例では、現存するシンクライアント・システムのサーバを修正する必要はないし、UniIntサーバ2をサポートするwindowシステム上で動作するあらゆるアプリケーションを、修正を行わずにシステムで制御することが可能である。

【0032】プロキシ手段としての汎用相互通信プロキシに相当するUniIntプロキシ3は、内部ネットワークとインターネットの境にあって、直接インターネットに接続できない内部ネットワークの機器に代わって、インターネットとの接続を代理して行なうもので、ネットワークに出入りするアクセスを一元管理し、内部から特定の種類の接続のみを許可したり、外部からの不正なアクセスを遮断するために用いられるが、これはとりわけ本システムにおいて最も重要な構成要素である。UniIntプロキシ3は、UniIntサーバ2から受け取ったビットマップ・イメージを、出力装置5の特性に従って変換する。また、UniIntプロキシ3は、入力装置4から受け取ったイベントを、UniIntサーバ2との間の汎用相互通信プロトコルに適應するマウス若しくはキーボードのイベントへと変換する。UniIntプロキシ3は、各機器を制御するために、現在の適正な入力/出力相互通信装置4,5を選択する。次に、選択された入力装置4は入力プラグイン・モジュールの信号を送信すると共に、選定された出力装置5は出力プラグイン・モジュールの信号をUniIntプロキシ3に送信する。入力プラグイン・モジュールは、入力装置4から受け取ったイベントやマウスやキーボードのイベントへ翻訳するコードを備えている。出力プラグイン・モジュールは、UniIntサーバ2から受け取ったビットマップ・イメージを、目標となる出力装置5の画面上に表示することのできるイメージに変換するコードを備えている。

【0033】入力装置4は、利用者との相互通信をサポートするもので、家庭用機器を制御するための利用者により発せられたコマンドを伝送するものである。出力装置5は、機器を制御するのに必要なグラフィカル・ユーザー・インターフェースを表示するための表示手段(表示装置)を備えている。

【0034】UniIntプロキシ3は、異種の相互通信装置4,5を扱う役目を演じる。また、UniIntプロキシ3は、利用者の状況や好みに応じて相互通信装置4,5の切り替えも可能である。これにより、利用者との間を独自のものとすることができる。

【0035】ここでのUniIntプロキシ3はJava言語によ

り書かれており、実施例では、図1に示す如く4つのモジュール部を備えている。第1のモジュールは、汎用相互通信プロトコルモジュール11であり、これはUniIntサーバ2と通信を行なうための汎用相互通信プロトコルを実行するものである。この第1のモジュール11を置き換えると、様々なシンクライアント・システムを備えたシステムを使用することができる。このモジュール11は、シンクライアント・システムの観点にて実行される同様なモジュールを利用できる。第2のモジュールは、プラグイン管理モジュール12で、これは相互通信装置4,5から入力/出力プラグイン・モジュールを受け取ると共に、UniIntプロキシ3内の各モジュールとダイナミックにリンク(連結)するものである。第3のモジュールは、利用可能装置検出手段に相当するプラグ・アンド・プレイ管理モジュール13で、これは現在利用可能な入力/出力相互通信装置4,5を検出するものである。第4のモジュールは、プラグイン移動管理モジュール14で、これは相互通信装置4,5とUniIntプロキシ3との間の入力/出力プラグイン・モジュールの移動を管理するものである。

【0036】プラグ・アンド・プレイ管理モジュール13は、UniIntプロキシ3の近くにある現在利用可能な入力装置4や出力装置5を検出する。ここでのシステムでは、独自のIDが各型式の入力装置4や出力装置5に割り当てられる。UniIntプロキシ3は、周期的にビーコン・メッセージを放送する。現行のプロトタイプ(代表例)では、相互通信装置4,5は、IEEE802.11b, Ethernet(登録商標)或いは赤外線ネットワークを通じて接続される。各相互通信装置4,5がビーコン・メッセージを受け取ると、そこから受信メッセージを回答する。受信メッセージは、その機器の型式を確認する独自のIDを含んでいる。仮にUniIntプロキシ3が、複数の相互通信装置4,5からそれぞれの受信メッセージを受け取ると、UniIntプロキシ3は、UniIntプロキシ3によって定められた選択に従って一つの相互通信装置4,5を選定する。また、新規に検出された相互通信装置4,5が受信メッセージを回答すると、例えその装置が現在使用されている装置より好ましいものであれば、その装置は現在使用中の相互通信装置4,5として選択される。

【0037】UniIntプロキシ3が、その相互通信装置4,5の検出後に新たな相互通信装置4,5を選択する場合は、新たな相互通信装置4,5の使用前に確認メッセージを送る。それからUniIntプロキシ3は、現在まで使われている相互通信装置4,5へ最終メッセージを送る。最後にUniIntプロキシ3は、新たな相互通信装置4,5からのプラグイン・モジュールの受け取りを待つ。入力装置4や出力装置5の選定は、各利用者用のUniIntプロキシ3中に登録されている。仮にシステムが機器を利用したい者を検出できない場合は、優先して不履行(デフォルト)が選択される。また、プラグイン・モ

ジュールのそれぞれは、現在使用中の入力装置4や出力装置5を切り替えるイベントを支援する。例えば、現在使用されている出力装置5を変更するために、利用者はUniIntプロキシ3に対しコマンドを送ることができる。UniIntプロキシ3は、利用者が自分の好みの出力装置5を選択するまでは、現在の出力装置4を次の出力装置5に切り替える。

【0038】UniIntプロキシ3からの受信メッセージを受け取ると、入力装置4や出力装置5は、プラグイン・モジュールをUniIntプロキシ3に送る。本システムでは、入力装置4や出力装置5とUniIntプロキシ3と間のプラグイン・モジュールを送信するために、MobileSpace移動体システムを使用している。UniIntプロキシ3内で実行されるJavaの仮想機器内にプラグイン・モジュールがダウンロードされた後、プラグイン移動モジュール14は、入力/出力装置4, 5へ移動完成メッセージを送る。MobileSpaceシステムは階層的エージェント(代理人)をサポートしており、同一階層レベルに存在する場合に、エージェントは通信を行なうことができる。それ故に、利用者はまたモバイルエージェントとしてUniIntプロキシ3を実行するが、そのエージェントは別のホストへ移動されない。しかしながら、その特徴は、入力/出力装置4, 5から近くのコンピュータへUniIntプロキシ3を移動させるのに利用してもよい。

【0039】ここでは、現存するJava言語で書かれたモバイルエージェントシステムを使用することを想定しているが、幾つかの制限のために装置の中にはJava仮想機械(VM)を持つことができないものもあるので、総ての装置においてJavaを使うという仮定は限界があると考えられる。その場合、エージェントを他のコンピュータへ送る為に、C言語で書かれた非常に小さなランタイム(アプリケーションソフトを実行する際に必要となるソフトウェアモジュール(部品))を構築することを考慮してもよい。このランタイムは、エージェントを実行できないが、別のランタイムへエージェントを送信できる。この様な小さなランタイム機能部は、種々の相互通信装置4, 5を支援する為に、望ましいものであると考える。また、この手法により、豊富な機能を提供するJavaに基づいたモバイルエージェントシステムを、一般的に採用することができる。このことは、非常に異なる要求を持つ如何なる装置に置いても使用可能な、標準的なモバイルエージェントシステムを構築することが可能であることを意味する。

【0040】図2は、プラグイン管理モジュール13, 14の機能構成を示すものである。このモジュール13, 14は、入力プラグイン・モジュール21と出力プラグイン・モジュール22を含んでいる。入力管理手段としての入力プラグイン・モジュール21は、現在選択されている入力装置4からイベントを受け取るものである。このイベントは、マウス・イベント若しくはキーボード・イベント

に変換され、変換されたイベントは、汎用相互通信プロトコルによって、UniIntサーバ2へ伝送される。例えば、利用者が右側の矢を描くボタンに触れると、そのイベントは入力装置4であるPDA(携帯情報端末)装置から発信され、入力プラグイン・モジュール21へと送信される。送信されたイベントは、右へのマウスの移動イベントへと翻訳され、そのイベントは最終的にwindowシステムに転送される。

【0041】同様に、UniIntプロキシ3内の汎用相互通信プロトコル・モジュール11により、UniIntサーバ2からビットマップ・イメージを受け取った後、出力管理手段としての出力プラグイン・モジュール22は、出力装置5へ伝送する前に、そのビットマップ・イメージを処理する。例えば、UniIntサーバ2から受け取った色彩イメージは白黒イメージに変換される。また、画像サイズは、出力装置5であるPDA装置の画面に縮小表示される。

【0042】本システムでは、2つの出力プラグイン・モジュール22と3つの入力プラグイン・モジュール21を供給する。第1の出力プラグイン・モジュール22Aは、標準のVGA画面にビットマップ・イメージを描くもので、第2の出力プラグイン・モジュール22BはPalm Pilot用である。また3つの入力プラグイン・モジュール21A, 22B, 22Cは、それぞれキーボードおよびマウス, Palm Pilot, コンパクトHTMLを支援するウェブ・ブラウザを有する携帯電話用である。

【0043】ホーム・コンピューティング・アプリケーション1の役割は、利用者が家庭用機器を制御できるようにすることである。利用者とは家庭用機器間の相互通信は、機器の機能が益々豊富になるので、より複雑になるであろう。また機器の数は、将来増加するであろう。それ故に、将来の家庭用機器は、種々の機器との状況認知("context-aware")相互通信をサポートしなければならない。

【0044】考慮すべき状況認知は、2つの型がある。一番目は、相互通信の個別化である。相互通信は、また利用者の状況に応じてカスタマイズされる。二番目の型は、一個の機器として現在利用できる複数の機器を処理することである。

【0045】システムは誰が機器を制御するかを知らないで、機器との相互通信を個別化することは通常困難である。本実施例におけるシステムでは、各利用者がPDA装置や携帯電話などのような自身の保有する制御装置を有することを仮定している。こうした制御装置が、利用者の確認IDを伝送するならば、システムは誰がその機器を制御したいかが判る。しかし本システムにおいては、利用者の確認をサポートしない従来のユーザー・インターフェース・システムをアプリケーション1が採用すると仮定しているため、相互通信装置4, 5からホーム・コンピューティング・アプリケーション1へ、その

様な情報を伝送する直接の方法は存在しない。それ故に、現在の実施例では、各利用者用に個別化されたアプリケーション1を実行する異なるUniIntサーバ2を、各利用者毎に備えている。アプリケーション1は、利用者の好みに応じてカスタマイズされたユーザー・インターフェースを供給する。UniIntプロキシ3は、入力装置4から得られる利用者の確認に応じて、最も相応しいUniIntサーバ2を選択する。

【0046】本実施例におけるシステムは、現在の状況によってどの機器が現在利用できるかを知る必要がある。現在のシステムにおいては、そのアプリケーション1をどの機器が利用できるのかを知っているものと仮定する。例えば、仮にアプリケーション1が3つの家庭用機器をサポートしているならば、そのアプリケーション1は、3つの機器を結合する7個のグラフィカル・ユーザー・インターフェースを提供が必要である。ユーザー・インターフェースは、現在利用可能な機器に応じて選択される。本システムでは、IEEE1394ネットワークを通じて各家庭用機器が接続されるものと仮定する。IEEE1394ネットワークは、どの機器が現在接続されているか、誰のパワースイッチがオンになっているかを知らせる機構をサポートしているので、アプリケーション1が現在利用可能な機器を容易に知ることができると共に、最も相応しいユーザー・インターフェースを容易に選ぶことができる。

【0047】一方、入力装置4や出力装置5は種々のものが利用可能であり、入力装置4と出力装置5を分離しても、あるいは結合してもよい。例えば、グラフィカル・ユーザー・インターフェースは、PDA装置の画面上或いはTV上の大型表示装置上に表示できる。TV上に表示されるユーザー・インターフェースは、PDA装置により操作することもある。それ故に、利用者は自分の好みによって相互通信方式の多様な選択ができる。またこれらの装置は、現在の状況に応じて変更が可能である。例えば、現在使用中の装置が利用できないならば、別の相互通信装置を機器制御のために選択してもよい。

【0048】本システムでは、前述したようにプラグイン・モジュールをUniIntプロキシ3へ送信すると仮定する。しかしながら、マイクロフォンの様なある入力装置4はプログラム化できないし、UniIntプロキシ3への送信をサポートすることは困難である。この場合は、我々は、その装置4を制御処理部であるパソコン(パーソナルコンピュータ)へ接続し、パソコンはUniIntプロキシ3と通信して、プラグイン・モジュールを送信する。しかしながら、パソコンはマイクロフォンが現在利用可能か否かを理解できないので、パソコン・メッセージがUniIntプロキシ3から受け取られた場合に、パソコン上のプログラムが確認メッセージを何時戻すのかを知ることが困難である。それ故に、現在のプロトタイプにおいては、その装置4が常に接続され利用可能であると仮定す

る。

【0049】次に、上記構成におけるシステムの動作を説明する。アプリケーション1が、現在利用可能な機器が例えばテレビジョンとビデオ・レコーダー(VCR)であると認識した時に、それらを制御するのに必要なグラフィカル・ユーザー・インターフェースを提示する。現在の利用者がそのTVプログラムの予約に興味がないとすると、アプリケーション1は、電源制御、TVチャンネルの選択、VCR機能を含むユーザー・インターフェースをUniIntサーバ2に描く。UniIntプロキシ3は、これ等の機器を制御したい利用者用にカスタマイズされたユーザー・インターフェースを描くアプリケーションを実行するUniIntサーバ2を選択すると共に、その選択されたUniIntサーバ2は、インターフェースを含むビットマップ・イメージをUniIntプロキシ3へ送信する。

【0050】ここで、利用者がPDA装置を備えていることを、UniIntプロキシ3が検出したと仮定する。入力/出力装置4,5に相当するPDA装置は、入力/出力プラグイン・モジュールの信号をUniIntプロキシ3に伝送する。UniIntプロキシ3は、ビットマップ・イメージをPDA装置へ送信する前に、出力プラグイン・モジュール22を使って、UniIntサーバ2から送信されたビットマップ・イメージをコード変換する。この場合、画像サイズが640×480の8ビットカラー映像は、画像サイズが、180×120の白黒画像に縮小される。また、PDA装置のタッチ・スクリーン上の入力イベントは、入力プラグイン・モジュール21によって、マウスとキーボード・イベントへ変換される。しかしながら、今、利用者は、グラフィカル・ユーザー・インターフェースを大画面で見たいものと仮定しよう。利用者は、PDA装置のタッチ・スクリーン上を軽く叩いて、UniIntプロキシ3へコマンドを送信する。UniIntプロキシ3がそれを検出すると、グラフィカル・ユーザー・インターフェースを含むビットマップ・イメージは、表示システムへ転送されると共に、ビットマップ・イメージを送信する前に、そのビットマップ・イメージは、大画面の表示システムに与えられるように、出力プラグイン・モジュール22によって変換される。またユーザー・インターフェースは、PDA装置のタッチ・スクリーン上を軽く叩くことにより、このタッチ・スクリーン上に再び表示される。

【0051】本実施例におけるシステムでは、種々の家庭用機器を制御するために、新型の携帯装置を使用できる。利用者が、眼鏡処方用レンズと区別つかないヘッド・マウント・ディスプレイを装着して、テレビジョンを制御したいものと仮定しよう。この場合、テレビジョンのグラフィカル・ユーザー・インターフェイスが、テレビジョンの近くにある眼鏡上に表示される。利用者は、音声を通じてグラフィカル・ユーザー・インターフェースを操作する。

【0052】UniIntプロキシ3は、グラス上に表示する

のに適した映像サイズに変換する。もしも、利用者が眼鏡を外して非装着状態になると、グラフィカル・ユーザー・インターフェイスは、自動的にテレビジョンの画面中に表示される。また、グラフィカル・ユーザー・インターフェイス上のカーソルを移動させるのにも、音声を利用される。音声はUniIntプロキシ3内のキーボードとマウスのイベントに翻訳され、これ等のイベントは、テレビジョン内で実行されるアプリケーション1に伝送される。

【0053】次に、本実施例におけるプロトタイプ・システムの現状を説明し、それから、ネットワーク化された家庭用機器を制御するためプロトタイプ・システム構築に関する幾つかの事例を説明する。

【0054】本実施例のシステムは、シンクライアント・システムとして、ネットワークに繋がった他のコンピュータの画面を遠隔操作するソフトウェアとして知られるVNC(Virtual Network Computing)システムを採用しているが、このVNCサーバは、UniIntサーバ3として修正せずに使用することができる。ここでのHAViに基づくホーム・コンピューティング・システムにおける現行プロトタイプは、2つの家庭用機器に機能させている。一番目はDVビューアで、二番目はデジタルTVエミュレーターである。ここでのアプリケーション1は、現在利用可能な機器に従ってグラフィカル・ユーザー・インターフェイスを表示する。また、グラフィカル・ユーザー・インターフェイスを表示する画面上のカーソルは、Palm Pilotから動かすことができる。しかしながら、装置の電源をオフしたら、グラフィカル・ユーザー・インターフェイス上のカーソルは、パソコンのカーソルとマウスによって制御される。利用者の好みに応じて、PDA装置上でグラフィカル・ユーザー・インターフェイスを表示することも可能である。また、現在のシステムは、家庭用機器を制御するために携帯電話と一体化されている。携帯電話の中には、ウェブ・ブラウザを有するものがある。これにより、携帯電話に表示される特別リンクをクリックすることで、カーソルの移動を可能にする。

【0055】図3は、実際のシステムの使用状態を示している。この図にも示すように、例えばデジタルビデオカメラとデジタルTVチューナが同時に利用できるなら、それらのためのコントロールパネルが、一枚のパネルとして結合される。この図では、表示装置21に表示されるコントロールパネルが、Palm pilotなどの携帯情報端末22と移動可能なモバイルパソコン23の両方により制御される。本実施例におけるホーム・コンピューティング・システムでは、資源(リソース)保存能力を追加することで、標準Linuxを拡大するLinux/Rtを採用した。Linuxはまた、IEEE1394装置ドライバーとMPEG2デコーダを提供する。また、Java仮想機器用のIBM社製JDK1.1.8が、HAViミドルウェア・コンポーネントを実行するために使用される。

【0056】本実施例のシステムでは、グラフィカル・ユーザー・インターフェイスを含むビットマップ・イメージが、UniIntサーバ2からUniIntプロキシ3へ送信される。ビットマップ・イメージは、自身の内容に関する意味情報を含んでいないので、UniIntプロキシ3は、その内容を理解できない。例えば、ビットマップ・イメージから各GUI(グラフィカル・ユーザー・インターフェイス)成分のレイアウトを引き出すことは困難である。それ故に、出力装置5の特性や利用者の好みに従ってレイアウトを変更することは容易ではない。また、本実施例のシステムは、マウスとキーボードのイベントによってのみ処理できる。このため、グラフィカル・ユーザー・インターフェイスの操作は、カーソルの移動とマウス・ボタンを押すことにより実行される。もしも、この限界がシステムの有用性を損なうならば、他の手法を選択すべきである。しかし、グラフィカル・ユーザー・インターフェイスをPDAと携帯電話から操作することにより、家庭用機器との非常に柔軟性のある相互通信ができる。家庭用機器は、一般的に大表示画面の使用を可能にしてくれると共に、各機器を制御するために画面上にグラフィカル・ユーザー・インターフェイスを表示することを可能にする。この様に、本実施例におけるシステムは、将来の家庭用機器用ミドルウェア・コンポーネントに柔軟性を持たせるのに十分な能力を有している。

【0057】また、本実施例におけるシステムは、X windowの様な標準のwindowシステム上で実行される如何なるアプリケーションをも制御できる。ここでのホーム・コンピューティング・システムにおいては、従来のアプリケーションがホーム・コンピューティング・アプリケーションと共存し、これらのアプリケーションは本システムによって、統一化された方法で制御可能である。例えば、本システムによって、ホーム・コンピューティング・アプリケーションと共に動作するMP3プレイヤーやNetscape(登録商標)ブラウザを操作することができる。しかしながら、重複するwindowレイアウトは、本実施例におけるユーザー・インターフェイス・システムによって操作することが難しい。家庭用機器を制御するには、いわゆる"tiled window strategy"が適していると考える。また、本実施例におけるシステムが、種々の速度でのマウス・カーソルの移動をサポートするならば、システムを楽な方法で使用するにより、家庭用機器と、プレゼンテーション・ソフトウェアやウェブ・ブラウザの様な従来のアプリケーションの両方を制御できる。

【0058】本実施例におけるホーム・コンピューティング・システムのアプリケーションは、状況情報に応じてユーザー・インターフェイスをカスタマイズするのに非常に簡単な機能を提供する。本システムでは、IEEE1394のプラグ・アンド・プレイ能力を使ってどの機器が使用可能かを監視する。また、利用できる機器の総ての

結合を知り、且つ各々を結合するためのユーザー・インターフェースを設計することが必要であると考え。それ故に、現在の構成に従って、それに相応しいユーザー・インターフェースが選定される。

【0059】プロトタイプのアプリケーションは、本システムの効果を証明するのに有用である。特に、各々の機器における機能構成と、利用者の好みに応じたユーザー・インターフェースのカスタマイズ化は、ネットワークされた家庭用機器を制御するのに有効である。しかしながら、状況認知("context-aware")アプリケーションを構築するための設計手法は、非常に特定されたものであり、それ故に、状況認知アプリケーションを組織的な方法で構築する研究が必要で、場所の情報や利用者の感情の様な、多くの状況情報を監視する種々のセンサーを必要とする。また、例えば新しい機器がIEEE1394ネットワーク上に現れようとも、自動的に機器における複数機能を構成する技術を調べる必要がある。

【0060】こうした問題を解決するために、2つの問題を研究している。一番目の問題は、組織的な方法で状況認知アプリケーションを構築することである。本手法では、先ず、状況("context")を考慮に入れない基本プログラムを作成する。それから、徐々に増加させながら、状況処理する新しい関係を加える。このプログラム内で、一つの状況は一つの関係として考えられ、各関係は基本プログラムから明瞭に分離される。こうしたアスペクト志向のプログラミングは、この問題を解決する助けとなると思われる。二番目の問題は、各状況を厳格に指定することである。従来のシステムでは、状況の表示は特別な方法で扱われる。しかしながら、ポータブルな状況認知アプリケーションを構築する為に、状況の表示は標準化されるべきである。

【0061】なお、本実施例におけるシステムと類似する幾つかの手法がある。そうした5つのシステムを説明すると共に、本実施例におけるシステムと比較を行なう。

【0062】第一のシステムは、Pebbleシステムである。このPebbleシステムにより、世界で最も知られているPDAのPalmPilotを介して、MS Windows(登録商標)オペレーティング・システム用のデスクトップ・アプリケーションを制御することができる。例えば、デスクトップ上のカーソルは、PalmPilotのスクリーンに触れることにより、動かすことができる。このシステムは、本実施例で提案したプロトタイプに非常に近い。しかしながらPebbleシステムは、システムの利用性に焦点を置くので、その目標は異なる。一方、本実施例におけるシステムは、システムの構造に焦点を絞っており、種々の入力装置4を使って、出力表示装置上においてグラフィカル・ユーザー・インターフェースを操作できると共に、入力/出力相互通信装置4, 5は、利用者の好みに従って切り替え可能である。本実施例におけるシステムの柔軟

性は、ネットワーク化された家庭用機器を支援する為により適している。

【0063】二番目のシステムは、UIML(User Interface Makeup Language)システムで、これは高度な装置の独立性を持って、ユーザー・インターフェースの宣言的記述が可能なXML言語である。もしも、アプリケーションがUIML文書としてユーザー・インターフェースを書くとすると、その文書は、各入力/出力相互通信装置に従って翻訳される。例えば、Palm Pilotを入力/出力装置として使用するために、UIML文書はPalm Pilot上に提供される。またUIML文書は、音声相互通信を支援する為にVoice XML用に提供される。しかしながら、入力/出力装置が分離されている場合は、この手法を採用することは難しい。また、利用者の状況に応じて、これらの入力/出力装置をダイナミックに切り替えるサポートが存在しない。

【0064】三番目のシステムであるCUESシステムは、モバイル装置から種々の機器を制御する骨組みを提供する。このシステムにおいては、各機器は、モバイル装置へ送信されるJavaバイトコードを持っている。このコードは、グラフィカル・ユーザー・インターフェースを含み、モバイルコンピュータ上に表示される。利用者は、モバイルコンピュータ上のグラフィカル・ユーザー・インターフェースによって、機器を制御できる。この手法により、各機器に対して適正なグラフィカル・ユーザー・インターフェースを使用できる。しかしながら、この手法では、モバイルコンピュータは、Java AWT/Swingにより実行されるグラフィカル・ユーザー・インターフェースを表示する中間サイズの表示器を有し、指示装置とキーボードを持つべきであることを前提としている。また、この手法は、ユーザー・インターフェースのカスタマイズと相互通信装置のダイナミックな切り替えはサポートしていない。

【0065】四番目のシステムであるSimjalは、マルチ・モデル・インターフェースをサービスにサポートするミドルウェア構成要素である。Simjalは、種々のメディア・フォーマットを翻訳する。それぞれの翻訳機は単一の機能を有し、別の翻訳機への接続が可能である。Simjalにおいては、この接続はパスと呼ばれ、パスは要求を表す仕様に従って自動的に翻訳機の中でセットアップされる。本実施例で提案したシステムの手法においては、プラグイン・モジュールがモノリシック("monolithic")であり、各相互通信装置が独立して入力/出力モジュールを形成することが必要であるので、このSimjalの手法は、プラグイン・モジュールを構築するには有益である。

【0066】五番目のシステムであるHAViには、利用者との相互通信をサポートする2つの方法がある。一番目の方法は、"Data Driven Interaction"(DDI)である。これは、グラフィカル・ユーザー・インターフェースを

記述するための宣言方法を提供する。この手法に於いて、ユーザー・インターフェースは、UIML の様な文書として記述され、この文書は表示装置の特性に応じて提供されるが、しかし、UIMLはXMLに基づいていて、HAViのDDIより一般的である。二番目の方法は、前記CUESシステムに似ている。Java AWTで書かれたユーザー・インターフェースを含むJavaバイトコードがHAVi装置にダウンロードされ、グラフィカル・ユーザー・インターフェースが装置の画面上に表示される。ここで生じる問題は、CUESの場合と同じである。

【0067】本実施例では、従来のグラフィカル・ユーザー・インターフェース・システムと、進歩したネットワーク化されたホーム・コンピューティング用入力/出力相互通信装置との間の隙間を埋める新しいユーザー・インターフェースについて説明してきた。実施例に提示するシステムは、グラフィカル・ユーザー・インターフェースを含むビットマップ・イメージの内容は分析しない。それ故に、ビットマップ・イメージが出力装置5へ翻訳されなければ、或いは、入力イベントがマウスまたはキーボードへ翻訳されなければ、本システムを利用するのは難しい。しかし、ネットワーク化された音響映像家庭用装置は、大画面の表示装置と共に使用されるのが普通なので、本システムが、これ等の家庭用機器を制御するには十分であると、我々は信じるものである。さらに本システムが、WindowsやLinux上で実行する種々のアプリケーションを制御するのに利用することができる。

【0068】また本実施例では、上述の隙間を埋めるユーザー・インターフェース・システムであるネットワーク化された家庭用機器用の全般的相互通信を提案した。このシステムは如何なる場所でも様な方法で機器の制御が可能であり、Java AWTまたはGTK+の様な従来の標準グラフィカル・ユーザー・インターフェイスを、アプリケーションが利用することができるが、利用者は、PDAや携帯電話などの種々の装置、若しくは進歩した技術により、そのインターフェイスを操作することができる。しかも、Citrix(登録商標) Metaframe, Microsoft(登録商標) Terminal Servier, Sun Microsystems Sun RayそしてAT&T(登録商標) VNC(VirtualNetwork Computing)などの国籍のないシンクライアント・システムに基づいた目標を、非常に容易に実現できる。そして、プロトタイプ・システムを構築し、利用者が自分の携帯する種々の相互通信装置を使用することができる。ここでのプロトタイプ・システムは、家庭用機器用の広く分布した標準ミドルウェア仕様であるHAViを導入したホーム・コンピューティング・システムと現在統合されており、これは家庭用機器を制御するのに有益である。

【0069】以上のように本実施例によれば、利用できる機器に応じて、その機器を制御するのに必要なグラフィカル・ユーザー・インターフェースを生成して提示す

るアプリケーション手段たるアプリケーション1と、前記機器を制御するためのコマンドを伝送する入力装置4と、表示手段を備えた出力装置5と、入力装置4および出力装置5と、アプリケーション1との間に介在するプロキシ手段としてのUniIntプロキシ3とを備え、UniIntプロキシ3は、入力装置4や出力装置5から受信メッセージを受け取ることにより、現在利用可能な入力装置4や出力装置5を検出してそれを選択する利用可能装置検出手段としてのプラグ・アンド・プレイ管理モジュール13と、アプリケーション1から汎用相互通信プロトコルによって送られてきたグラフィカル・ユーザー・インターフェースに含まれるイメージを、プラグ・アンド・プレイ管理モジュール13が選択した出力装置5の表示手段に表示できるように変換し、この変換したイメージを含むグラフィカル・ユーザー・インターフェースを前記出力装置5の表示手段に転送する出力管理手段としての出力プラグイン・モジュール22と、プラグ・アンド・プレイ管理モジュール13が選択した入力装置4からコマンドを受取ると、このコマンドをアプリケーション1が認識できるイベントに変換し、この変換したイベントを汎用相互通信プロトコルによってアプリケーション1へ伝送する入力管理手段としての入力プラグイン・モジュール21とを備えて構成される。

【0070】この場合、利用可能な機器に応じたグラフィカル・ユーザー・インターフェースが、アプリケーション1から汎用相互通信プロトコルによってUniIntプロキシ3に送られると、そのグラフィカル・ユーザー・インターフェースに含まれるイメージは、UniIntプロキシ3により現在利用可能な出力装置5の表示手段に適合するように変換され、その変換されたグラフィカル・ユーザー・インターフェースが表示手段に転送される。また、利用者が利用可能な機器を制御するには、表示手段に表示されるグラフィカル・ユーザー・インターフェースを参照しながら、現在利用できる入力装置4から適宜コマンドを伝送すればよい。こうすれば、どのような入力装置4であっても、UniIntプロキシ3は伝送されたコマンドを、汎用相互通信プロトコルに適合する例えばマウスまたはキーボードのイベントに変換し、それをアプリケーション1に伝送する。

【0071】つまり、ここでのUniIntプロキシ3はいわば異種の入力装置4や出力装置5を扱う手段としての機能を果すので、現存する標準のグラフィカル・ユーザー・インターフェースを変更することなく、種々の入力装置4や出力装置5を使って、様な手順で機器の制御を行なうことができる。

【0072】また、本実施例では、入力装置4から所望の出力装置5を選択するコマンドを受取ると、プラグ・アンド・プレイ管理モジュール13による選択に優先して、この出力装置5の表示手段が表示できるグラフィカル・ユーザー・インターフェースを出力装置5に転送す

るように、出力プラグイン・モジュール22を構成している。

【0073】このようにすると、入力装置4から表示したい出力装置5を選択するコマンドを伝送すると、その出力装置5の表示手段に適合したグラフィカル・ユーザー・インターフェースがUniIntプロキシ3から送り出される。したがって、利用者の状況や好みに応じた出力装置5の切替えが可能となり、利用者と機器との間を独自のものとする事ができる。

【0074】さらに本実施例では、アプリケーション1が現在利用可能な機器を特定するために、IEEE1394ネットワークを通じてこれらの機器と接続している。

【0075】IEEE1394ネットワークは、どの機器が現在接続されているか、誰のパワースイッチがオンになっているかを知らせる機構をサポートしているので、アプリケーション1が現在利用可能な機器を容易に知ることができると共に、最も相応しいユーザー・インターフェースを容易に選定することができる。

【0076】なお、本発明は上記実施例に限定されることなく、本発明の範囲内で種々の変形実施が可能である。

【0077】

【発明の効果】本発明の請求項1の構成によれば、利用者がいかなる場所にいたとしても、従来の標準グラフィ

カル・ユーザー・インターフェースをそのまま利用して、一様な手順で機器の制御を行なうことができる。

【0078】本発明の請求項2の構成によれば、利用者の状況や好みに応じた出力装置の切替えが可能になる。

【0079】本発明の請求項3の構成によれば、現在利用可能な機器を容易に知ることができると共に、最も相応しいユーザー・インターフェースを容易に選定できる。

【図面の簡単な説明】

【図1】本発明の一実施例を示すシステムの全体構成をあらわした概略説明図である。

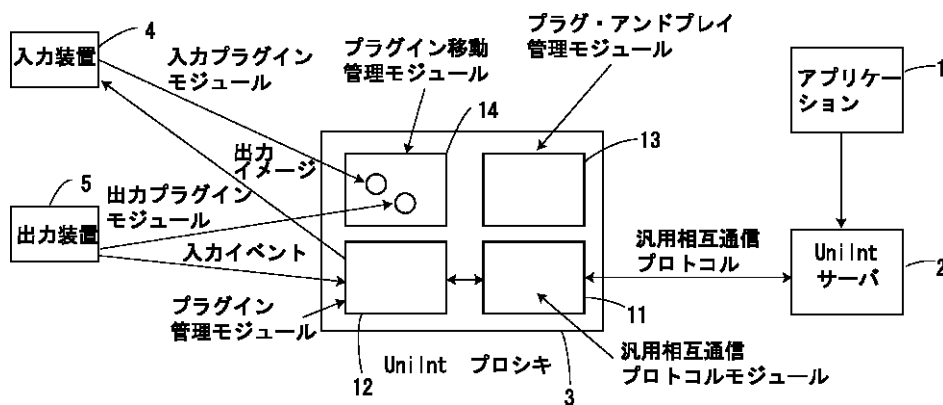
【図2】同上プラグイン管理モジュールの機能構成を示す概略説明図である。

【図3】同上実際のシステムの使用状態を示す概略説明図である。

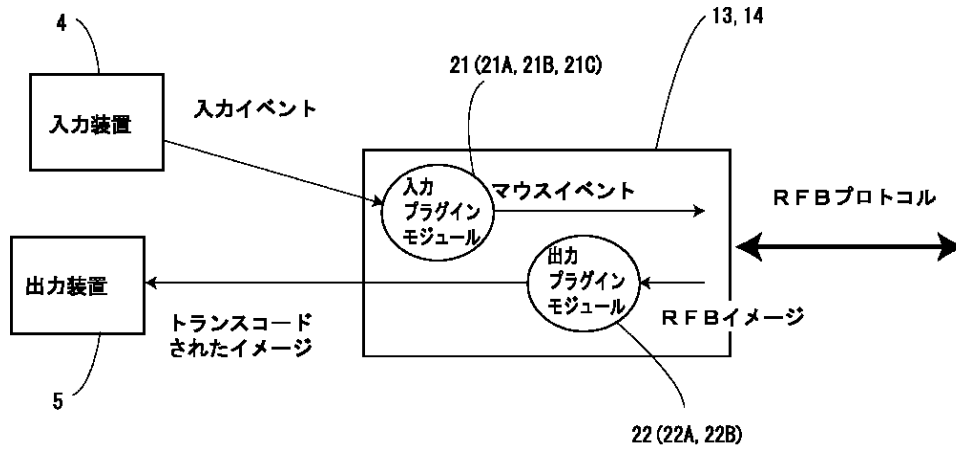
【符号の説明】

- 1 アプリケーション（アプリケーション手段）
- 3 UniIntプロキシ（プロキシ手段）
- 4 入力装置
- 5 出力装置
- 13 プラグ・アンド・プレイ管理モジュール（利用可能装置検出手段）
- 21 入力プラグイン・モジュール（入力管理手段）
- 22 出力プラグイン・モジュール（出力管理手段）

【図1】



【図2】



【図3】

