US 20020062406A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2002/0062406 A1

Chang et al. (43) Pub. Date: May 23, 2002

(54) OUTPUT DEVICE FOR MOBILE AND PERVASIVE OUTPUT

(76) Inventors: **William Ho Chang**, Vancouver, WA (US); **Christina Ying Liu**, Vancouver, WA (US)

Correspondence Address:
IPSOLON LLP
805 SW BROADWAY, #2740
PORTLAND, OR 97205 (US)

(21) Appl. No.: **09/992,183**

(22) Filed: **Nov. 18, 2001**

Related U.S. Application Data

(63) Non-provisional of provisional application No. 60/252,682, filed on Nov. 20, 2000.

Publication Classification

(51) **Int. Cl.**[7] **G06F 3/00**

(52) **U.S. Cl.** **710/1**

(57) **ABSTRACT**

Pervasive output capability is provided so that an information apparatus can pervasively output digital content to an output device regardless of the processing power, display screen size and memory space of the information apparatus. This allows small mobile devices with limited display, processing power, and memory to be able to output a digital document (e.g., PDF, HTML, PowerPoint etc) in its full original richness, without resorting to downsizing, truncating, reducing, clipping or otherwise altering the original document. A data output device for rendering output content managed from an information apparatus may include a rendering engine that imparts output on a medium in accordance with output data acceptable to the output device and a memory storage that stores at least part of one or more output device objects, with one or more attributes corresponding to the output device. The output device may further include means for providing at least part of the output device object to the information apparatus and means for receiving output data from the information apparatus, the output data related to the output content managed from the information apparatus and acceptable to the output device.

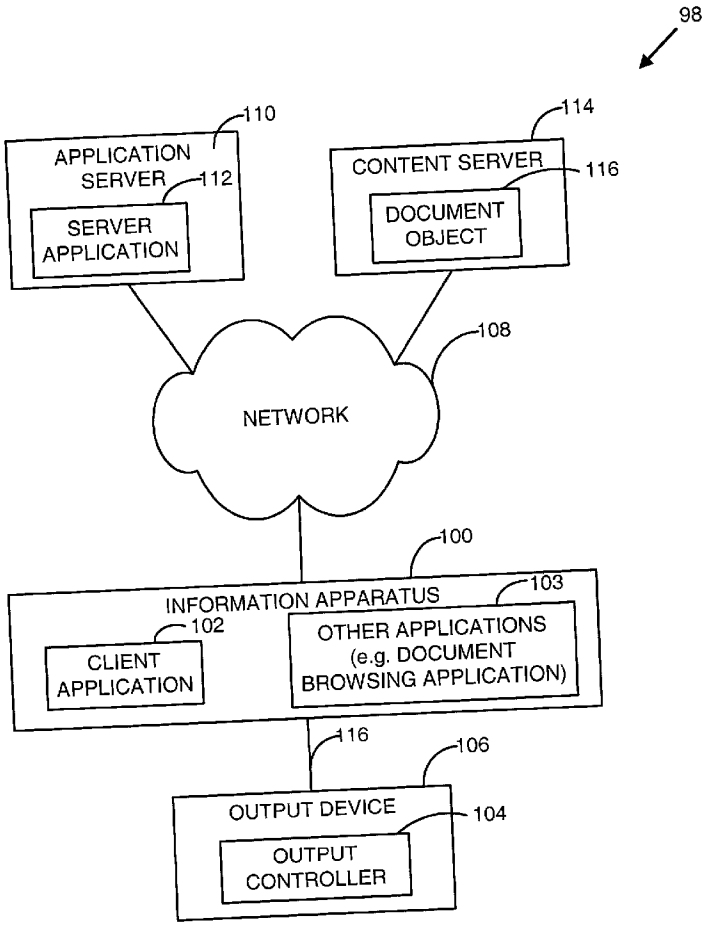
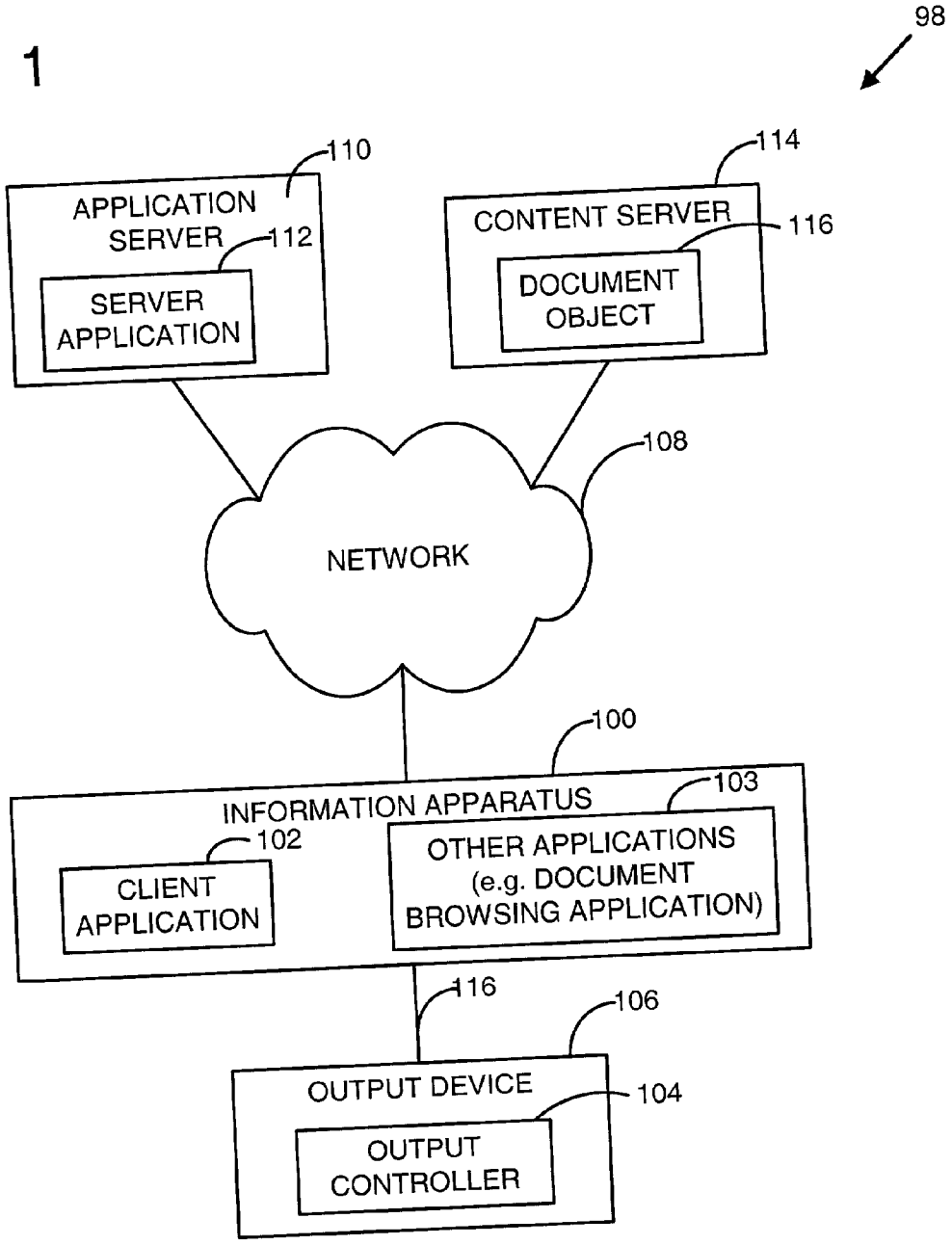


Fig. 1


98
110
APPLICATION SERVER
112
SERVER APPLICATION
114
CONTENT SERVER
116
DOCUMENT OBJECT
108
NETWORK
100
INFORMATION APPARATUS
103
102
CLIENT APPLICATION
OTHER APPLICATIONS (e.g. DOCUMENT BROWSING APPLICATION)
116
106
OUTPUT DEVICE
104
OUTPUT CONTROLLER

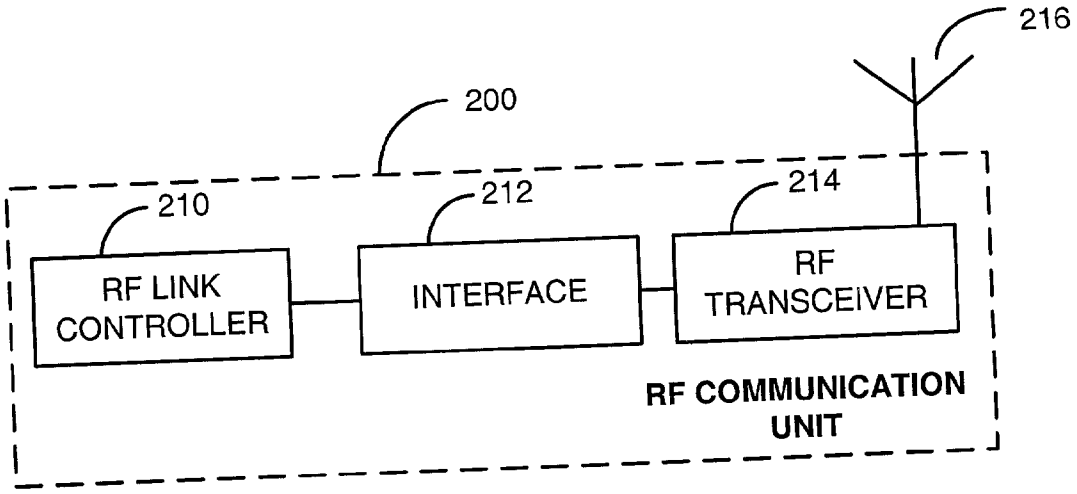

216
200
210
212
214
RF LINK CONTROLLER
INTERFACE
RF TRANSCEIVER
RF COMMUNICATION UNIT

Fig. 2A

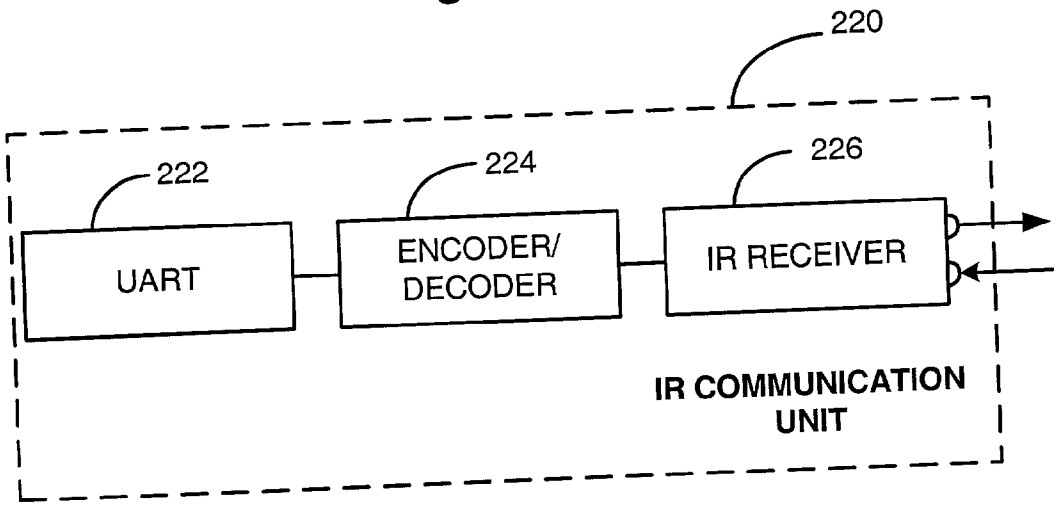

220
222
224
226
UART
ENCODER/ DECODER
IR RECEIVER
IR COMMUNICATION UNIT

Fig. 2B

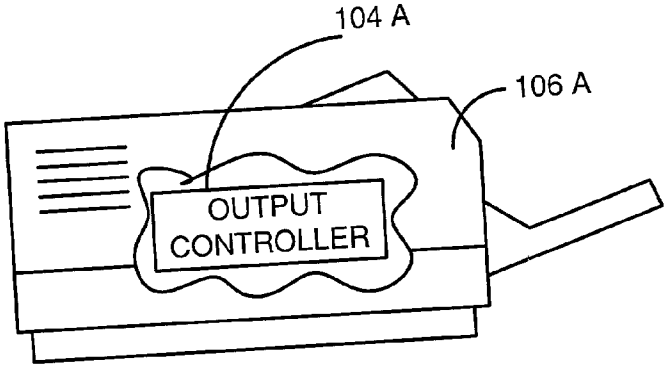


Fig. 3A

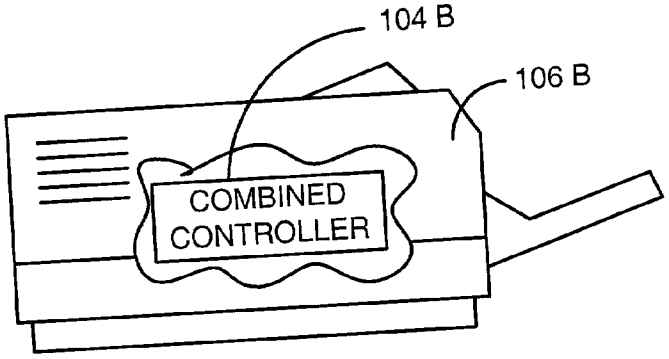


Fig. 3B

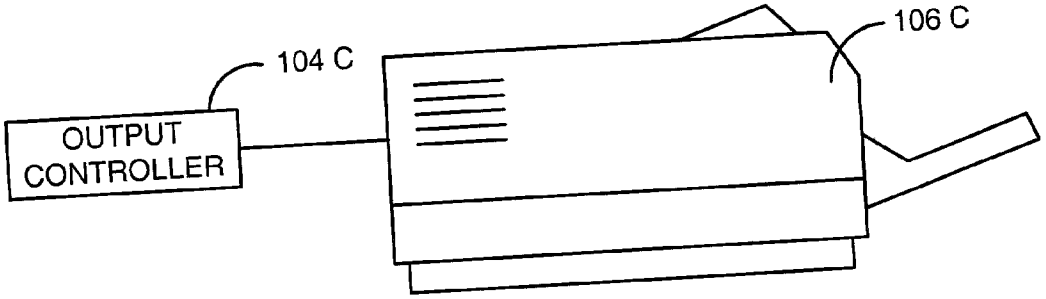


Fig. 3C

Fig. 4

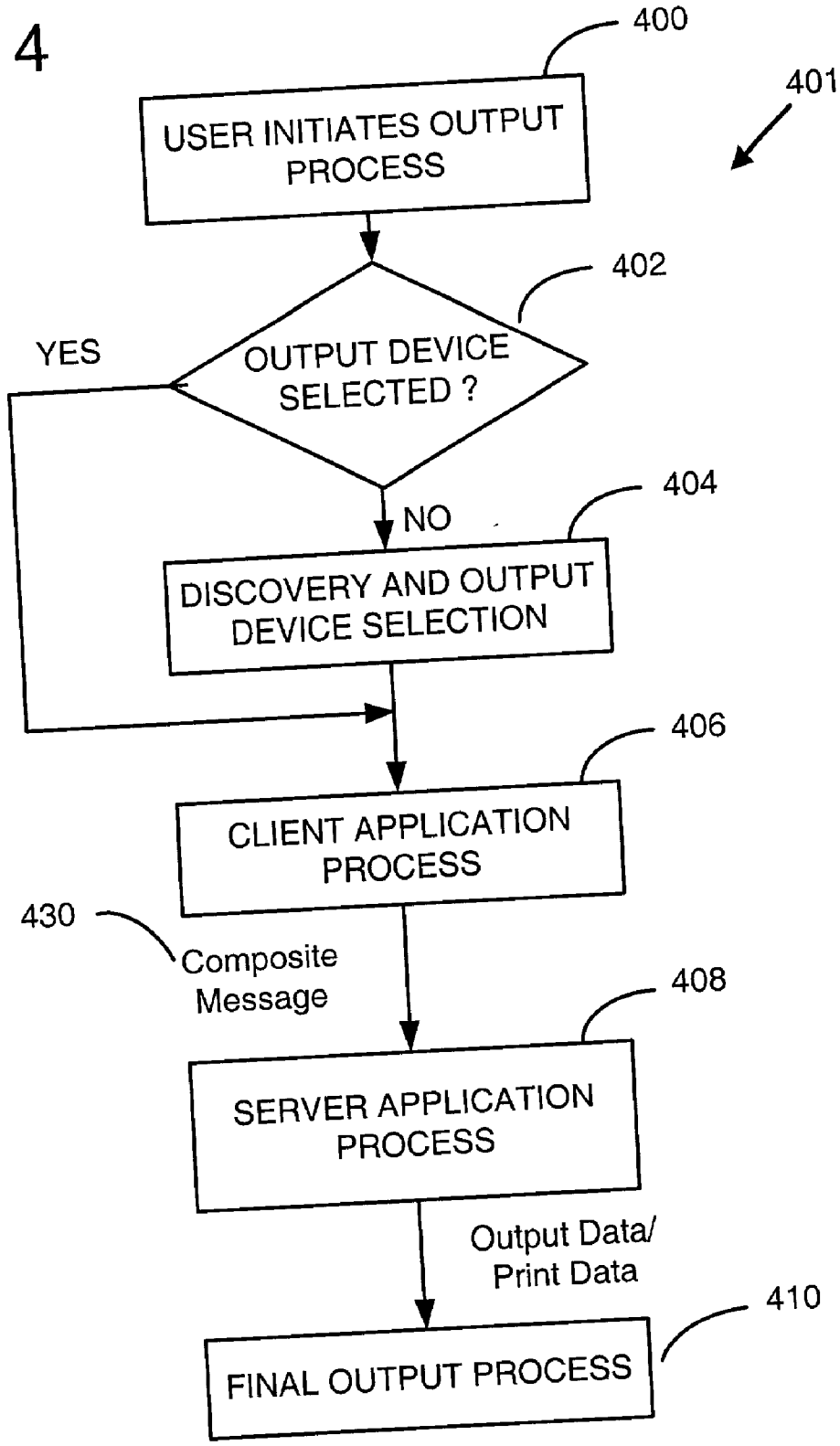

400
401
USER INITIATES OUTPUT PROCESS
402
OUTPUT DEVICE SELECTED ?
YES
NO
404
DISCOVERY AND OUTPUT DEVICE SELECTION
406
CLIENT APPLICATION PROCESS
430
Composite Message
408
SERVER APPLICATION PROCESS
Output Data/ Print Data
410
FINAL OUTPUT PROCESS

Fig. 5

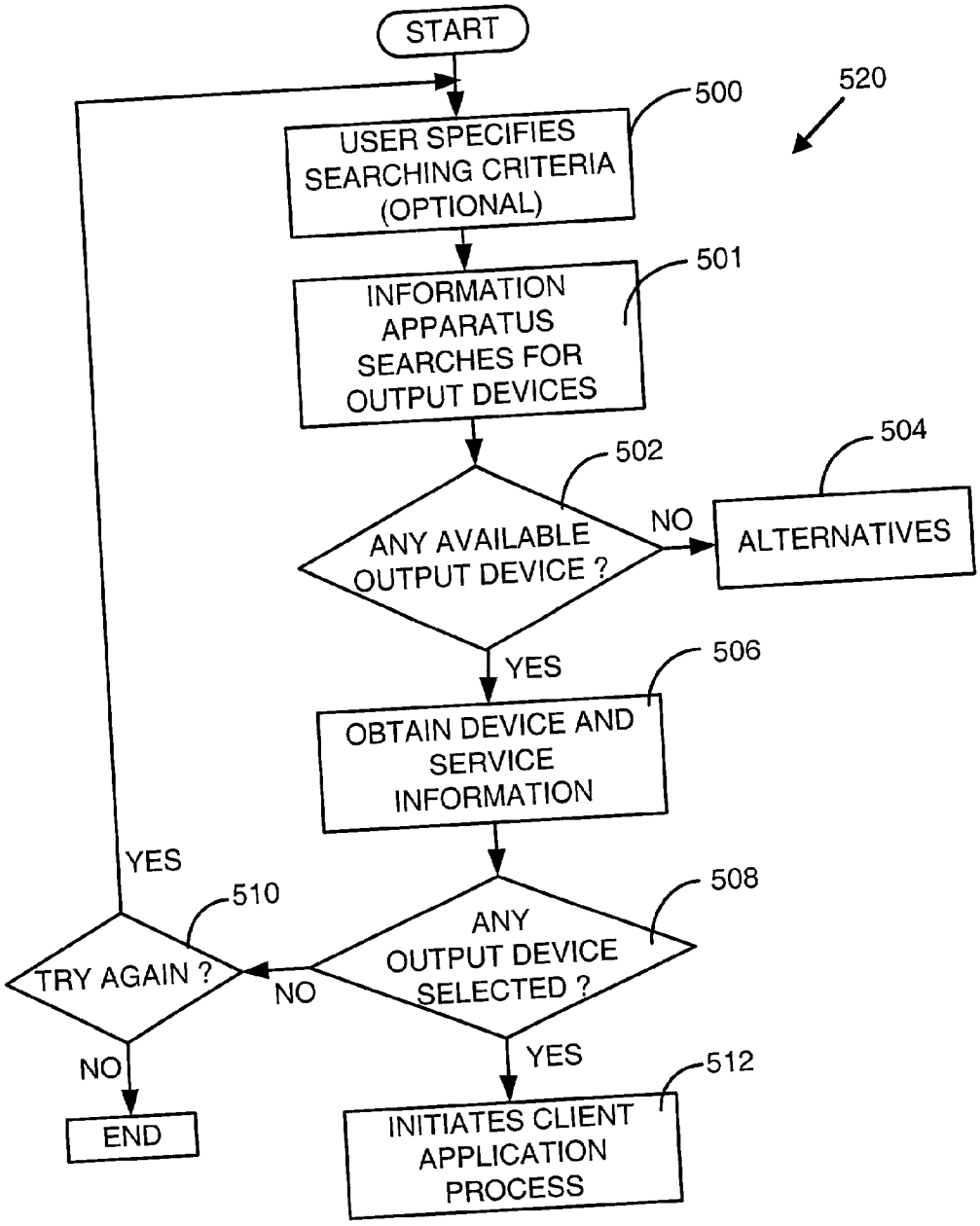
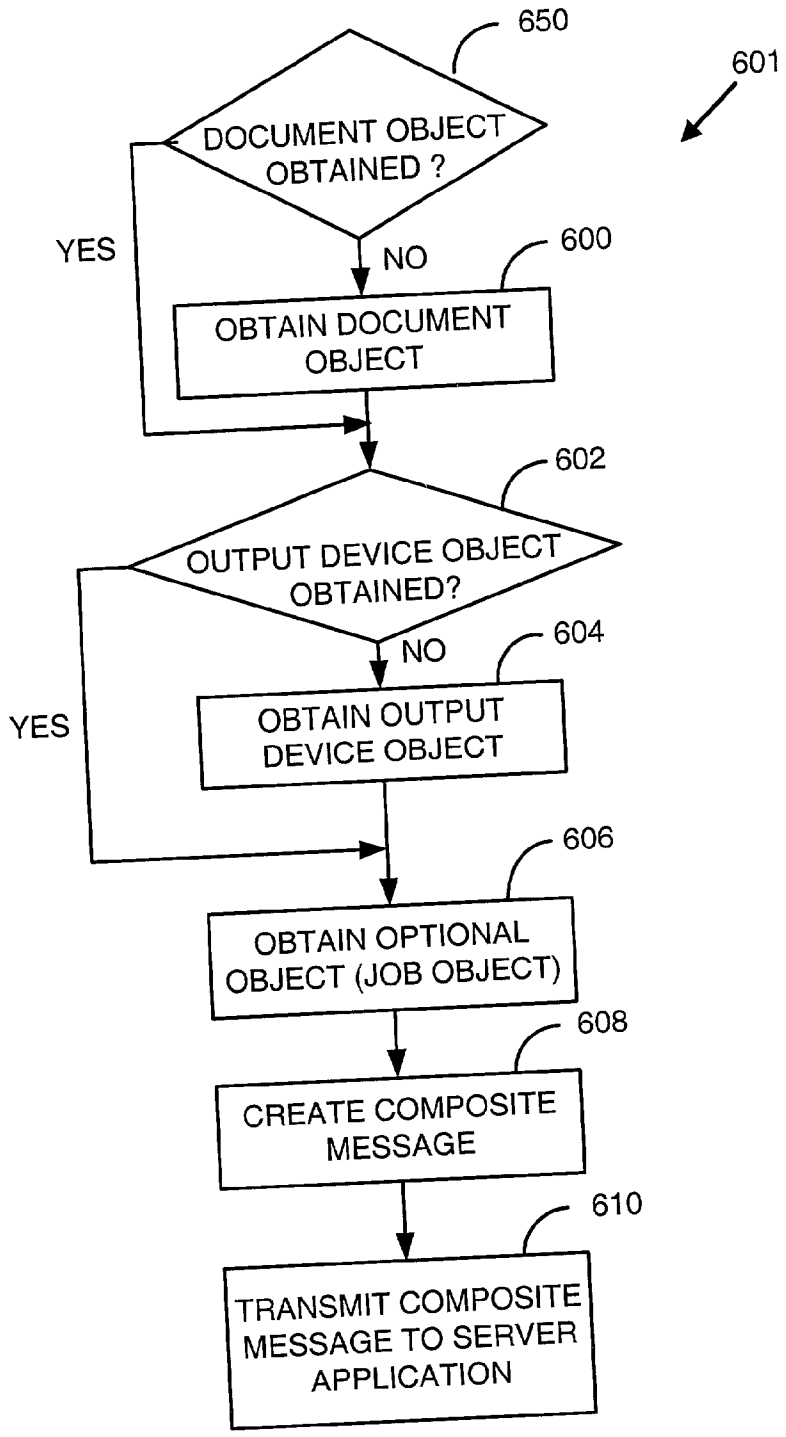

START
520
500
USER SPECIFIES SEARCHING CRITERIA (OPTIONAL)
501
INFORMATION APPARATUS SEARCHES FOR OUTPUT DEVICES
502
504
ANY AVAILABLE OUTPUT DEVICE ?
NO
ALTERNATIVES
YES
506
OBTAIN DEVICE AND SERVICE INFORMATION
YES
510
508
TRY AGAIN ?
NO
ANY OUTPUT DEVICE SELECTED ?
NO
YES
512
END
INITIATES CLIENT APPLICATION PROCESS

Fig. 6


650
601
DOCUMENT OBJECT OBTAINED ?
YES
NO
600
OBTAIN DOCUMENT OBJECT
602
OUTPUT DEVICE OBJECT OBTAINED?
NO
604
YES
OBTAIN OUTPUT DEVICE OBJECT
606
OBTAIN OPTIONAL OBJECT (JOB OBJECT)
608
CREATE COMPOSITE MESSAGE
610
TRANSMIT COMPOSITE MESSAGE TO SERVER APPLICATION

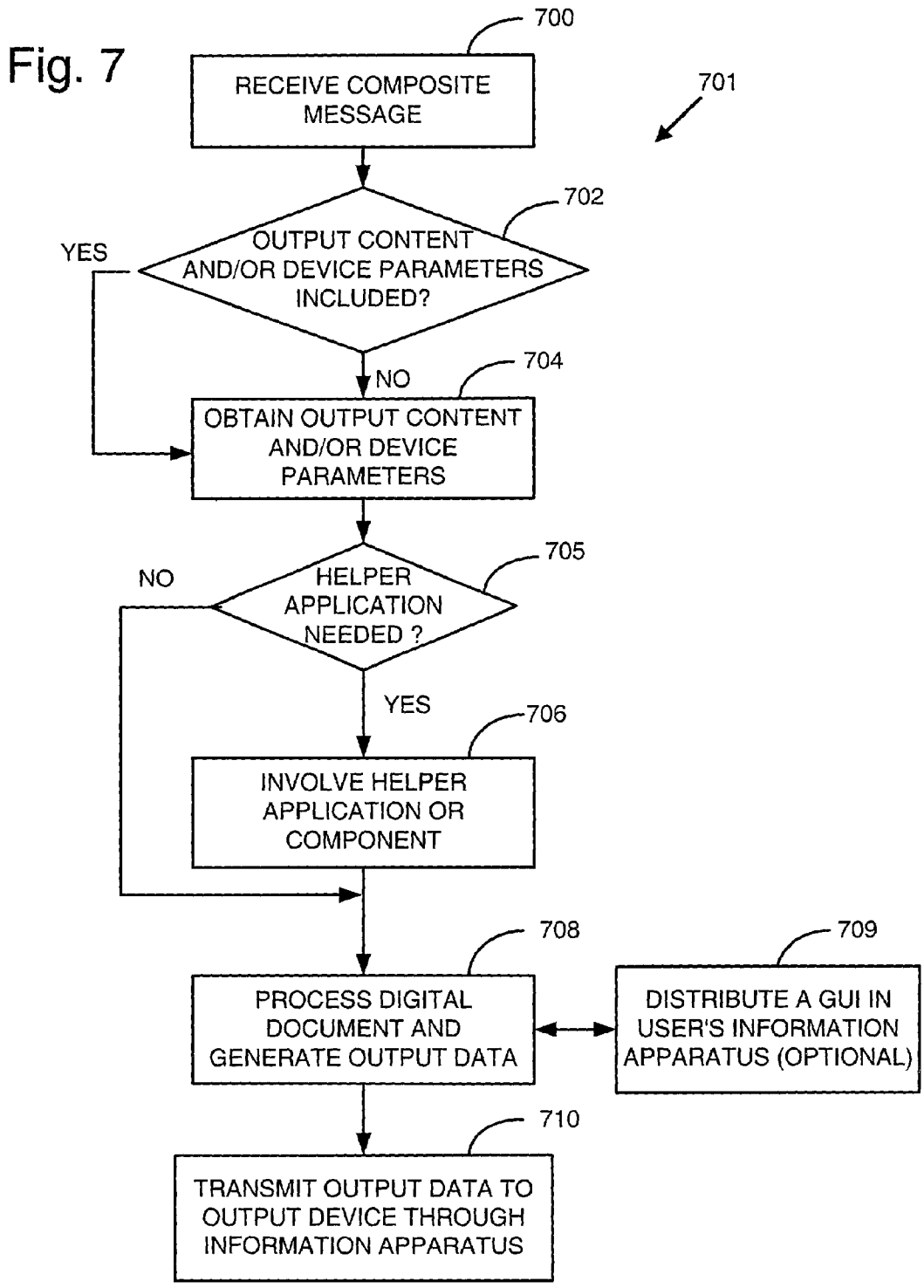
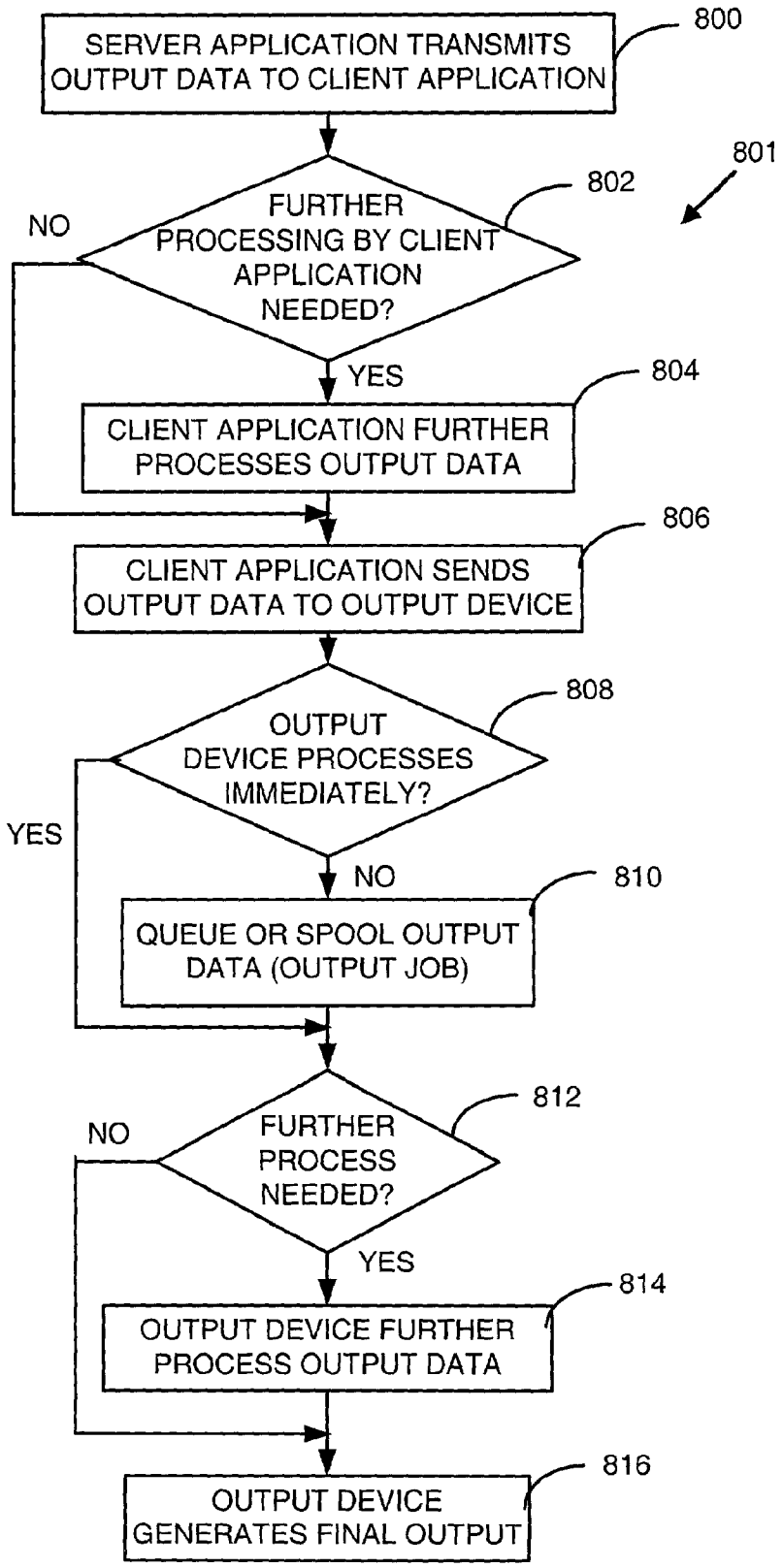


Fig. 7
700
RECEIVE COMPOSITE MESSAGE
701
702
OUTPUT CONTENT AND/OR DEVICE PARAMETERS INCLUDED?
YES
NO
704
OBTAIN OUTPUT CONTENT AND/OR DEVICE PARAMETERS
705
HELPER APPLICATION NEEDED ?
NO
YES
706
INVOLVE HELPER APPLICATION OR COMPONENT
708
PROCESS DIGITAL DOCUMENT AND GENERATE OUTPUT DATA
709
DISTRIBUTE A GUI IN USER'S INFORMATION APPARATUS (OPTIONAL)
710
TRANSMIT OUTPUT DATA TO OUTPUT DEVICE THROUGH INFORMATION APPARATUS

Fig. 8


SERVER APPLICATION TRANSMITS OUTPUT DATA TO CLIENT APPLICATION
800
801
FURTHER PROCESSING BY CLIENT APPLICATION NEEDED?
802
NO
YES
CLIENT APPLICATION FURTHER PROCESSES OUTPUT DATA
804
CLIENT APPLICATION SENDS OUTPUT DATA TO OUTPUT DEVICE
806
OUTPUT DEVICE PROCESSES IMMEDIATELY?
808
YES
NO
QUEUE OR SPOOL OUTPUT DATA (OUTPUT JOB)
810
FURTHER PROCESS NEEDED?
812
NO
YES
OUTPUT DEVICE FURTHER PROCESS OUTPUT DATA
814
OUTPUT DEVICE GENERATES FINAL OUTPUT
816

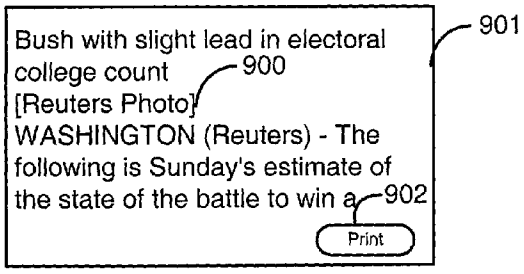


Fig. 9 A

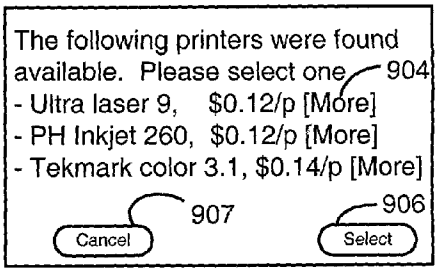


Fig. 9 B

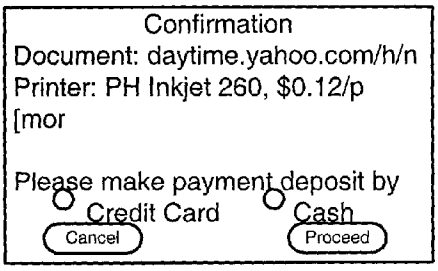


Fig. 9 C

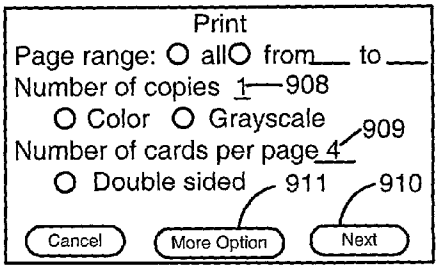


Fig. 9 D

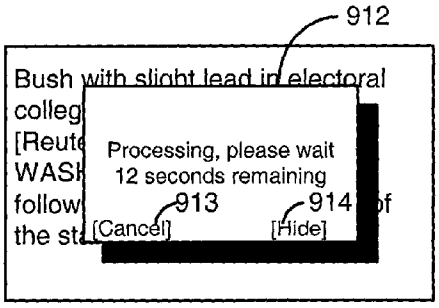


Fig. 9 E

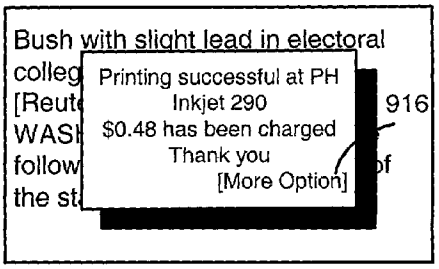


Fig. 9 F

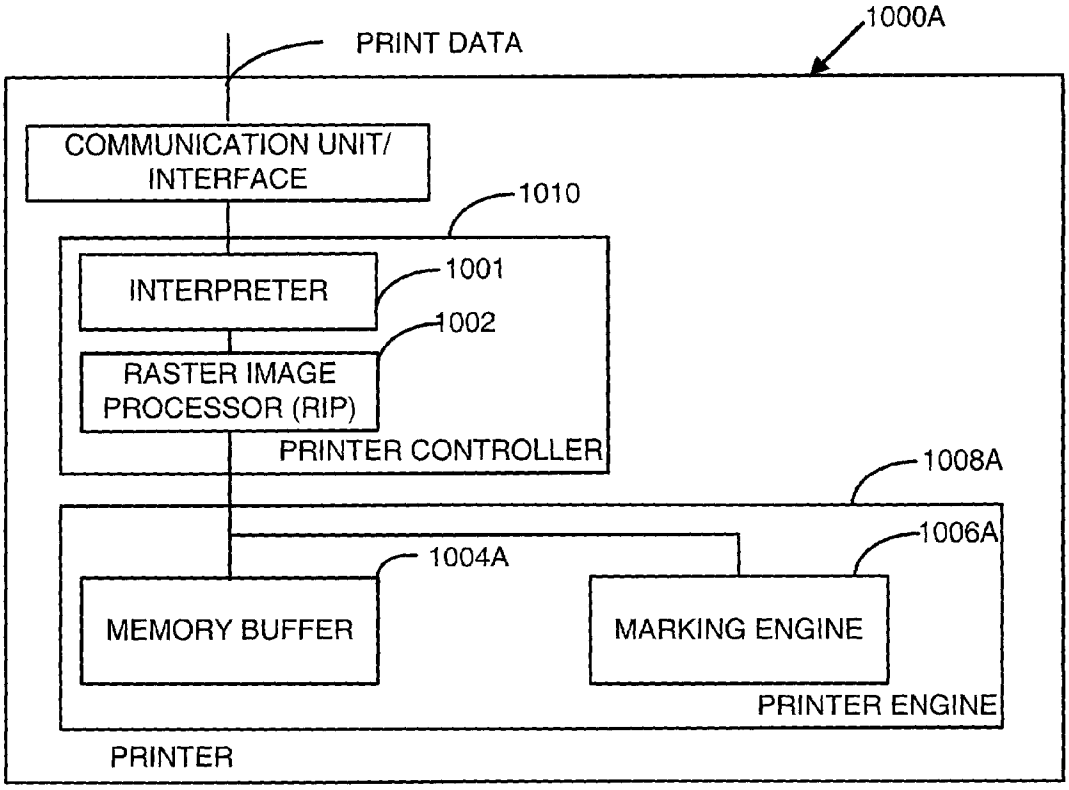


Fig. 10 A

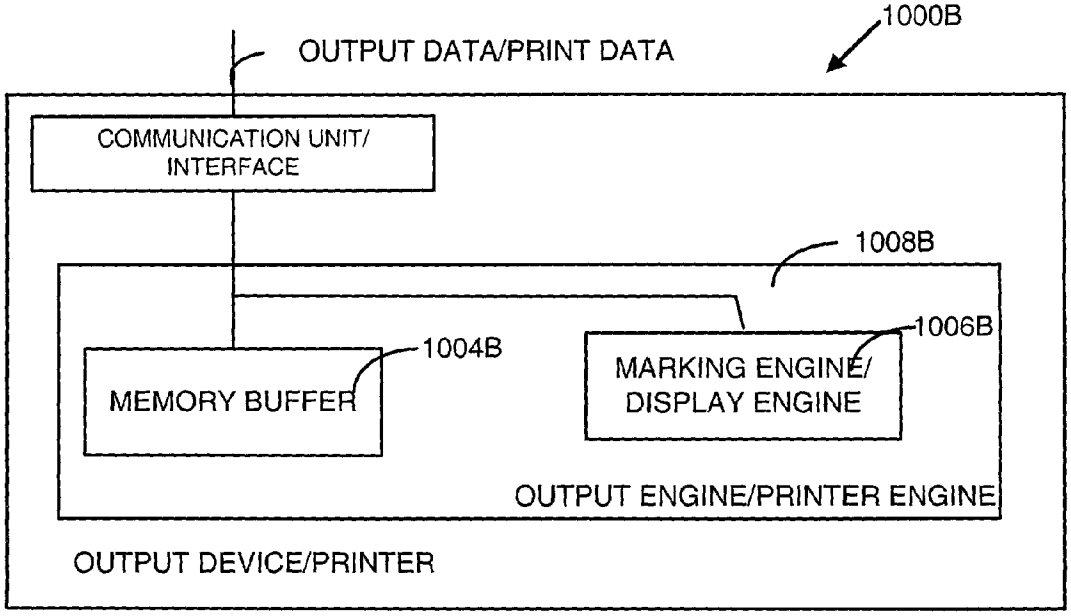


Fig. 10 B

OUTPUT DEVICE FOR MOBILE AND PERVASIVE OUTPUT

TECHNICAL FIELD OF THE INVENTION

[0001] Present invention relates to providing digital content to an output device and, in particular, to providing pervasive output in which an information apparatus can pervasively output digital content to an output device regardless of the processing power, display screen size and memory space of the information apparatus.

BACKGROUND AND SUMMARY OF THE INVENTION

[0002] As described herein, information apparatuses refer generally to computing devices, which include both stationary computers and mobile computing devices (pervasive devices). Examples of such information apparatuses include, without limitation, desktop computers, laptop computers, networked computers, palmtop computers (hand-held computers), personal digital assistants (PDAs), Internet enabled mobile phones, smart phones, pagers, digital capturing devices (e.g., digital cameras and video cameras), Internet appliances, e-books, information pads, and digital or web pads. An output device may include any one or more of fax machines, printers, copiers, image and/or video display devices (e.g., televisions, monitors and projectors), and audio output devices.

[0003] For simplicity and convenience, hereafter, the following description may refer to an output device as a printer and an output process as printing. However, it should be understood that the term printer and printing used in the discussion of present invention may refer to a specific example used to simplify description or may be one exemplary embodiment. The reference to printer and printing used here is intended to be applied or extended to the larger scope and definition of output devices and should not be construed as restricting the scope and practice of present invention.

[0004] Fueled by ever-increasing bandwidth and processing power and ever-increasing numbers of wireless mobile devices and available software applications for pervasive devices, millions of users are or will be creating, downloading, and transmitting content and information using their pervasive computing devices. As a result, there is a need to allow users to easily output content and information from their pervasive computing devices to any output device. People need to output directly and conveniently from their pervasive information apparatus, without depending on synchronizing with a stationary computer (e.g., desktop personal computer) for printing, as an example.

[0005] To illustrate, an information worker at an airport receiving Email in his hand-held computer may want to walk up to a nearby printer or fax machine to have his e-mail printed. In addition, the mobile worker may also want to print a copy of his to-do list, appointment book, business card, and his flight schedule from his mobile device. As another example, a user reading a news article using his/her Internet-enabled pager or mobile phone may want to print out the complete article instead of reading it through the small screen on the pager or mobile device. In still another example, a user visiting an e-commerce site using his mobile phone may want to print out pictures of the product he/she is buying because the pictures (image and/or graphics) may not be displayed on the small screen of his/her mobile phone. In yet another example, a user who takes a picture with a digital camera may want to easily print it out to a nearby printer. In still another example, a user with a mobile device may want to simply walk up to a printer and conveniently print a file that is stored on the mobile device or that is stored on a network (e.g., Internet, corporate network) and accessible from the mobile device, such as a PowerPoint® display application document, word processing document, or a document in any other file format such as PDF, HTML, JPEG etc. In addition, a user should also be able to print a web page or a book or a report published on the Internet. Finally, a user may want to output a complete web page or any document or file to a larger display screen nearby, even though the small screen of his/her mobile device cannot completely display or open a document of such a size.

[0006] Conventionally, an output device (e.g., a printer) is connected to an information apparatus via a wired connection such as a cable line. A wireless connection is also possible by using, for example, radio communication or infrared communication. Regardless of wired or wireless connection, a user must first install in the information apparatus an output device driver (e.g., printer driver in the case the output device is a printer) corresponding to a particular output device model and make. Using a device-dependent or specific driver, the information apparatus may process output content or digital document into a specific output device's input space (e.g., printer input space). The output device input space corresponds to the type of input that an output device (e.g., a printer) understands (herein referred to as output data or print data in the case the output device is a printer). For example, the printer input space or print data may include printer specific input format (e.g., one or more of an image format, graphics format, text format, audio format, video format, file format, and data format), encoding, language (e.g., page description language, markup language etc), instructions, protocols or data that can be understood or used by a particular printer make and model.

[0007] Output data may be proprietary or published or a combination of the two. An output device's input space or output data is therefore, in general, device dependent. Different output device models may have their own input spaces specified, designed or adopted by the output device manufacturer (e.g., the printer manufacturer) according to a specification for optimal operation. Consequently, different output devices usually require use of specific output device drivers (e.g., printer drivers) for accurate output (e.g., printing). For example, a printer driver may control, manage, communicate, and output print data to a printer. Sometimes, instead of using a device driver (e.g., printer driver), the device driving feature may be included or as part of an application software.

[0008] Installation of a device driver (e.g., printer driver) or application may be accomplished by, for example, manual installation using a CD or floppy disk supplied by the printer manufacturer. Or alternatively, a user may be able to download that particular driver or application from a network. For a home or office user, this installation process may take anywhere from several minutes to several hours depending on the type of driver and user's sophistication level with computing devices and networks. Even with plug-and-play

driver installation, the user is still required to execute a multi-step process for each printer or output device.

[0009] This installation and configuration process adds a degree of complexity and work to end-users who may otherwise spend their time doing other productive or enjoyable work. Moreover, many unsophisticated users may be discouraged from adding new peripherals (e.g., printers, scanners, etc.) to their home computers or networks to avoid the inconvenience of installation and configuration. Therefore, there is a need to provide a manner in which a user can more conveniently or easily output digital content to an output device without the inconvenience of finding and installing new device drivers or printer drivers.

[0010] In addition, conventional output or printing methods may pose significantly higher challenges and difficulties for mobile device users than for home and office users. The requirement for pre-installation of a device-dependent driver diminishes the benefit and concept of mobile (pervasive) computing and output. For example, a mobile user may want to print or output e-mail, PowerPoint® presentation documents, web pages, or other documents in an airplane or at an airport, gas station, convenient store, kiosk, hotel, conference room, office, home, etc. It is highly unlikely that the user would find at any of these locations a printer of the same make and model as is at the user's base station. It is usually not a viable option to preinstall all of the possible hundreds, or even thousands, of printer drivers or device drivers available to the user's information apparatus.

[0011] As a consequence, the user would currently have to install and configure a printer driver each time at each such remote location before printing. Moreover, the user may not want to be bothered with looking for a driver or downloading it and installing it just to print out or display one page of email at the airport. This is certainly an undesirable and discouraging process to promote pervasive or mobile computing. Therefore, a more convenient or automated printing and output solution is needed so that a user can simply walk up to an output device (e.g., printer or display device) and easily output a digital document without having to install or pre-install a particular output device driver (e.g., printer driver).

[0012] Another challenge for mobile users is that many mobile information apparatuses have limited memory space, processing capacity and power. These limitations are more apparent for small and low-cost mobile devices including, for example, PDAs, mobile phones, screen phones, pagers, e-books, Internet Pads, Internet appliances etc. Limited memory space poses difficulties in installing and running large or complete printer or device drivers, not to mention multiple drivers for a variety of printers and output devices. Slow processing speed and limited power supply create difficulties driving an output device. For example, processing or converting a digital document into output data by a small mobile information apparatus may be so slow that it is not suitable for productive output. Heavy processing may also drain or consume power or battery resources. Therefore, a method is needed so that a small mobile device, with limited processing capabilities, can still reasonably output digital content to various output devices.

[0013] Finally, some small mobile devices with limited display screens, such as mobile phones, may in some cases be limited to display only a few lines of text. Browsing the Internet with such devices can be a disappointing experience when viewing, for example, complex web pages containing rich formats, graphics, and images. Furthermore, some small mobile devices may not have appropriate applications to display complex documents or languages such as PDF-format files, word processing documents and PowerPoint® (D presentation documents etc. Typically, if an application is available, displaying complex original documents on small mobile devices may require downsizing the document or page into, for example, a few lines of text. As an example, WAP protocol, I-Mode, and web clipping among others may downsize, reduce or truncate information on the original web page for display on mobile devices. Therefore, it is desirable to allow mobile users to output from their small information apparatuses to an output device the full richness of the original document content.

[0014] One implementation of the present invention provides an easy, friendly and convenient process for digital output. Unlike conventional output or printing, a user does not have to manually pre-install a device driver (e.g., printer driver) from a CD, floppy disk, or download the driver somewhere from a network. This is well-suited for providing output capability to small and lower-cost mobile devices with limited memory space, power supply and processing capability to still be able to output or print to an output device.

[0015] In addition, this allows small mobile devices with limited display, processing power, and memory to be able to output a digital document (e.g., PDF, HTML, PowerPoint etc) in its full original richness, without resorting to downsizing, truncating, reducing, clipping or otherwise altering the original document. A user can output the original content or document even when the small mobile device cannot display or fully display the original digital document or content.

[0016] Finally, one implementation provides a convenient method allowing users to output to an output device with or without connection to a static network. Through local communication and synchronization between information apparatus and output device, hardware and software installation for static or permanent network connectivity may not be necessary for the output device.

[0017] Additional objects and advantages of the present invention will be apparent from the detailed description of the preferred embodiment thereof, which proceeds with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] **FIG. 1** is a block diagram of a pervasive output system that can implement the process and apparatus of the present invention.

[0019] **FIGS. 2A and 2B** are block diagrams illustrating exemplary configurations of hardware components of wireless communication units.

[0020] FIGS. **3A-3C** illustrate various configurations and implementations of output controller with respect to an output device such as a printer.

[0021] **FIG. 4** is a flow diagram of a pervasive output process of the present invention.

[0022] **FIG. 5** is a flow diagram of an exemplary implementation of a discovery process optionally included in the output process of **FIG. 4**.

[0023] **FIG. 6** is a flow diagram of an exemplary client application process included in the output process of **FIG. 4**.

[0024] **FIG. 7** is a flow diagram of an exemplary server application process utilized in the output process of **FIG. 4**.

[0025] **FIG. 8** is a flow diagram of an exemplary final output process for pervasive output.

[0026] FIGS. **9A-9F** show a series of exemplary graphical user interfaces (GUIs) rendered at different times on a display screen of an information apparatus.

[0027] **FIG. 10A** is an exemplary block diagram of a conventional printing system or printer with a conventional printer controller.

[0028] **FIG. 10B** is an exemplary block diagram of a conventional output system or an output device such as a printing system or printer that does not include a conventional printer controller.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0029] Sets forth below are definitions of terms that are used in describing implementations of the present invention. These definitions are provided to facilitate understanding and illustration of implementations of the present invention and should in no way be construed as limiting the scope of the invention to a particular example, class, or category.

[0030] Objects

[0031] An object may refer to a software and data entity, which may reside in different hardware environments or platforms or applications. An object may encapsulate within itself both data and attributes describing the object, as well as instructions for operating that data. For simplicity of discussion, an object may also include, for example, the concept of software components that may have varying granularity and can consist of one class, a composite of classes, or an entire application.

[0032] It is important to note that the term object is not limited to software or data as its media. Any entity containing information, descriptions, attributes, data, instructions etc. in any computer-readable form or medium such as hardware, software, files based on or including voice, text, graphics, image, or video information, electronic signals in analog or digital form, etc., are all valid forms of object definition.

[0033] An object may also contain in one of its fields or attributes a reference or pointer to another object, or a reference or pointer to data and or content. (The terms reference and pointer may be used interchangeably herein.) A reference to an object or any entity or content may include one or more, or a combination of, pointers, identifiers, names, paths, addresses or any descriptions relating to a location where an object, data, or content can be found. Examples of reference may include universal resource identifier scheme (URI), uniform resource locator (URL), IP address, file names, directory pointers, software object and component pointers, and run-time address, among others.

[0034] By way of example, a document object described in the present invention may contain or encapsulate one or more digital documents and/or one or more pointers or references to digital documents. Therefore, moving or passing document objects in connection with the present invention may include moving or passing (1) actual digital content or (2) reference to the actual content or (3) both. It will be appreciated that the document object can be quite small and lightweight if it does not also contain the digital document. These are examples of valid implementations and may be used in the description of present invention. Different implementations in different situations can be easily discerned and recognized by persons of ordinary skill in the art.

[0035] An object can reside anywhere in a network and can be transmitted to different environments, platforms, or applications. Downloading or transferring an object over the network may involve protocols such as file transfer protocol (FTP) or hypertext transfer protocol (http), among others. Transferring an object may also involve using messages or other methods through an object or component model. Three major objects used and described in present invention are output device object (or printer object in the case the output device is a printer), job object, and document object.

[0036] Output Device Object (or Printer Object in the Case the Output Device is a Printer)

[0037] An output device object may contain one or more attributes that may identify and describe, for example, the capabilities and functionalities of a particular output device such as a printer. An output device object may be stored in the memory component of an output device. As described below in greater detail, an information apparatus requesting output service may communicate with an output device. During such local service negotiation, at least a partial output device object may be uploaded to the information apparatus from the output device. By obtaining the output device object (or printer object in the case of a printer), the information apparatus may learn about the capability, compatibility, identification, and service provided by the output device.

[0038] As an example, an output device object or printer object may contain one or more of the following fields and or attribute descriptions. Each of following fields may be optional, and furthermore, each of the following fields or attributes may or may not exist in a particular implementation (e.g., may be empty or NULL).

[0039] Identification of an output device (e.g., brand, model, registration, IP address etc.)

[0040] Services and feature sets provided by an output device (e.g., color or grayscale output, laser or inkjet, duplex, output quality, price per page, quality of service, etc.)

[0041] The type of input languages, formats, and or output data (e.g., PostScript, PCL, XML, RTL, etc.) supported by an output device.

[0042] Device specific or dependent profiles (e.g., output device profiles, printer profiles, color profiles, halftoning profiles, communication profiles etc.). Device profiles may include information such as color tables, resolution, halftoning, dpi (dots-per-

inch), bit depth, page size, printing speed, etc. One or multiple profiles may exist.

[0043] Payment information on a plurality of services provided by an output device.

[0044] Information or security requirements and type of authentication an output device supports.

[0045] Date and version of the output device object (e.g., printer object).

[0046] Software components containing algorithms or instructions or data, which may be uploaded to run in an information apparatus, as described with reference to **FIG. 1**. For example, a graphical user interface (GUI) software component may be uploaded to an information apparatus. The software component may be incorporated into or launched in the information apparatus by a client application of present invention to capture a user's preferences (e.g., print quality, page layout, number of copies, number of card per page, etc.). The client application may capture these user preferences and create a job object, as described below.

[0047] Pointer or reference to any one or more output device parameters (output device parameters include one or more of the above described output device object fields and or attribute descriptions). For example, a more up-to-date or original version of output device parameters may sometimes be stored in a network node. An output device or may include pointer or pointers to these output device parameters.

[0048] Pointer or reference to another object or objects, including output device object, job object (described below) or document object.

[0049] Job Object

[0050] A job object may contain attributes and information that describe an output job. A user may provide some or all of these attributes, preferences and or information about the output job consciously by, for example, specifying his/her preference through a GUI or through defaults in his/her information apparatus. Alternatively or in combination, a job object may be obtained without active user intervention. In one instance, default parameters may be provided, obtained, negotiated or calculated without user knowledge.

[0051] Examples of attributes and information contained in a job object may include one or more of the following, among others. Each of following fields may be optional, and furthermore, each of the following fields or attributes may or may not exist in a particular implementation (e.g., may be empty or NULL):

[0052] Preferences such as print quality, page layout, number of pages, number of cards per page, output size, color or grayscale, among others.

[0053] Information on security, authentication, payment, subscription, identification among others.

[0054] Information on priority or quality of service.

[0055] Status information of the output job or process.

[0056] Job instructions such as queuing, cancellation, execution, output priority among others.

[0057] Version or date of the job object.

[0058] Sets of default parameters or instructions. The defaults may be stored in an output device, in the information apparatus, or in any network nodes.

[0059] Pointer or reference to any one of the above mentioned information, instructions, preferences and defaults.

[0060] Pointer or reference to another object or objects.

[0061] Document Object

[0062] A document object may contain attributes and fields that describe a digital document and or reference or references to digital document or documents. The term digital document as used herein may refer to any digital content or data content that an output device may output. A digital document may contain text, graphics, image, sound, voice, forms, and video, among other content types. Examples of a digital document may be any one or combination of file types: HTML, VHTML, PostScript, PCL, XML, PDF, MS Word, PowerPoint, JPEG, MPEG, GIF, PNG, WML, VWML, CHTML, HDML, ASCII, 2-byte international coded characters, etc. A digital document can be composed of any format, language, encoding, data or combination, and the digital document may be partially or totally proprietary or otherwise. A digital document may be used interchangeably with the term output content or data content in the descriptions of present invention.

[0063] A document object may contain one or more of the following attributes, fields, or descriptions. Each of the following fields may be optional, and furthermore, each of the following fields or attributes may or may not exist in a particular implementation (e.g., may be empty or NULL).

[0064] The actual digital document or output content.

[0065] A pointer or reference to a digital document or output content and or instructions of where a digital document can be found and or retrieved. A digital document may be located in a user's information apparatus, in an application, or anywhere in a network node (e.g., in a content or file server). Using a pointer or reference to a digital document may reduce the size of the document object. Therefore, this may be beneficial, for example, when passing or uploading a document object from information apparatus to server application through a narrow bandwidth communication link.

[0066] Date and version of the digital document.

[0067] A history with the dates and descriptions of any changes, modifications, and updates made to the document since its creation.

[0068] Descriptions and instructions for viewing, obtaining, opening, interpreting, encoding, decoding, compressing, decompressing, rendering, converting, describing, processing, and manipulating the digital document.

[0069] Status and state of the digital document. For example, one of the fields may provide priority or

instructions of when or where to use the reference to digital document or the actual digital document itself if they are both contained in a document object.

[0070] Pointer or reference to another object or objects.

[0071] **FIG. 1** is a block diagram of a pervasive output system **98** that can implement the process and apparatus of present invention. In one implementation, electronic system **98** includes an information apparatus **100**, an application server **110** and a content server **114** (sometimes referred to as "network nodes") that are connected together through network **108**. Electronic system **98** may also include an output device **106** that communicates with information apparatus **100** through a communication link **116**.

[0072] Network **108** generally refers to any type of wire or wireless link between multiple computing devices. Examples of network **108** may include, but are not limited to, a local area network (LAN), a wide area network (WAN), or a combination of networks. In one embodiment of the present invention, network **108** may include the Internet. In another embodiment, network **108** may contain multiple networks, including local area networks or wide area networks such as the Internet.

[0073] Information apparatus **100** is a computing device with processing capability. In one embodiment, information apparatus **100** may be a mobile computing device such as palmtop computer, handheld device, laptop computer, personal digital assistant (PDA), smart phone, screen phone, e-book, Internet pad, communication pad, Internet appliance, pager, digital camera, etc. It is possible that information apparatus **100** may also include a static computing device such as a desktop computer, workstation, server, etc.

[0074] Information apparatus **100** may contain components (not shown) such as a processing unit, a memory unit, a storage unit and an input/output control unit. Information apparatus **100** may also contain an interface (not shown) for interactions with users. The interface may be implemented in software or hardware or a combination. Examples of such interfaces include, without limitation, one or more of a mouse, a keyboard, a touch-sensitive or non-touch-sensitive screen, push buttons, soft keys, a stylus, a speaker, a microphone, etc.

[0075] Information apparatus **100** typically contains at least one network communication unit that interfaces with other electronic devices such as other nodes in network **108** or output device **106**. The network communication unit may be implemented with hardware (e.g., silicon chipsets, antenna), software (e.g., protocol stacks, applications) or a combination. Sometimes an information apparatus **100** may contain more than one communication unit in order to support different interfaces, protocols, and or communication standards with different devices and or network nodes. For example, information apparatus **100** illustrated in **FIG. 1** may communicate with output device **106** through a Bluetooth standard interface while communicating with other network nodes (e.g., content server **114** or application server **110**) through a cellular telephone modem interface.

[0076] Information apparatus **100** may be coupled to network **108** through wired or wireless connections, or a combination of them. As an example, information apparatus **100** may subscribe to a wireless data network in which packet data is transmitted through, for example, radio links between information apparatus **100** and a plurality of base stations. A wireless communication connection may include a cellular telephone communication channel. As another example, information apparatus **100** may be connected to network **108** through wired lines such as, without limitation, telephone lines, Ethernet, WAN links (e.g., T1, T3, 56 kb, X.25) or broadband connections (e.g., ISDN, Frame Relay and ATM), among others.

[0077] In one embodiment of present invention, interface **116** between information apparatus **100** and output device **106** is a wireless interface. As an example, the wireless interface may be a short-range radio interface such as those implemented according to the Bluetooth or IEEE 802.11 standard. However, the interface may be realized by other means of wireless communication such as radio, infrared, ultrasonic or hydrophonic among others. The HomeRF Shared Wireless Access Protocol (SWAP) and one of many standards for cellular communication may constitute alternatives to the Bluetooth standard. Wired line connections such as serial or parallel interface, USB interface and fire wire (IEEE 1394) interface, among others, are also possible. Connection to a local network such as an Ethernet or a token Ring network, among others, may also be implemented in the present invention for local communication between information apparatus **100** and output device **106**. Exemplary hardware components of communication units that may be used to implement wireless interface between the information apparatus **100** and output device **106** are described below with reference to **FIGS. 2A and 2B**.

[0078] Information apparatus **100** may be a dedicated device (e.g., email terminal, web terminal, digital camera, e-book, web pads, internet appliances etc.) with functionalities that are pre-configured by manufacturers. Alternatively, information apparatus **100** may allow users to install additional hardware components and or application software to expand its functionality.

[0079] Information apparatus **100** may contain a plurality of applications to implement its feature sets and functionalities. As an example, a document browsing application **103** may be implemented to help a user view and perhaps edit, partially or entirely, digital documents written in certain format or language (e.g., Page description language, markup language, etc.). Digital documents may be stored locally in the information apparatus **100** or in a network node (e.g., in content server **114**). An example of a document browsing application is an Internet browser such as Internet Explorer, Netscape Navigator, or a WAP browser. Such browsers may use one or more standard protocols (e.g., HTTP, WAP, web clipping, I-Mode, etc.) to retrieve and display digital content written in mark-up languages such as HTML, WML, XML, CHTML, HDML, among others. Other software applications may also constitute examples of the document browsing application of the present invention. For example, a document editing software such as Microsoft Word™ also allows users to view and edit digital documents that have various file extensions (e.g., doc, rtf, html, XML etc.) whether stored locally in the information apparatus or in a network node.

[0080] In some instances restrictions may be imposed on the format or size of digital content that may be transmitted to information apparatus **100**, such as when information

apparatus **100** has limited processing power, screen size, memory space, or a limited application, or when bandwidth is a valuable resource in the transmission link to information apparatus **100** (such as in some wireless data network). As a result, there are situations where a user may not be able to view on information apparatus **100** the full content of a digital document in its original form using a document browsing application. For example, some images, tables, graphics, fonts and formats in a digital document may be "clipped" out or completely or partially altered from the original content before or during the transmission process. Such restrictions may be, sometimes, imposed by a service providing the content, or by the application rendering the content, or by the user to avoid slow transmission etc.

[**0081**] To address the difficulties described above, information apparatus **100** includes a pervasive output client application **102** that provides pervasive output capability of the present invention. Client application **102** may include software and data that can be executed by the processing unit of information apparatus **100**. Client application **102** may be implemented as a stand-alone software application or as a part or feature of another application software, or in the form of device driver, which may be invoked, shared and used by other application software. Pervasive output client application **102** may also have means to invoke other applications (e.g., a document browsing application, a communication manager, etc.) to provide certain feature sets, as described below. Client application **102** may be variously implemented in an information apparatus **100** and may run on different operating systems or platforms. As an example, client application **102** may include one or more of the following functionalities:

[**0082**] Obtain output device object as a result of communication or negotiation with output device **106**.

[**0083**] Obtain document object (1) from user input or selection, or (2) from other applications (e.g., a document browsing application) residing in the information apparatus **100**.

[**0084**] Coordinate with a server application **112** residing in application server **110** to manage the process of communication and transmission of objects or data to and from application server **112**.

[**0085**] Coordinate with output device **106** that include an output controller **104** to manage the process of transmitting output data (or print data in the case of printers) received from the server application **112** for output.

[**0086**] The client application may also optionally comprise one or more of the following functionalities:

[**0087**] Communicate directly or indirectly (such as through an operating system or component or object model or message, etc.) with other applications residing in the same information apparatus **100** to obtain objects, data, and or content needed, or relating to the pervasive output process of present invention.

[**0088**] Directly or indirectly manage and utilize functionalities provided by hardware components residing in its host information apparatus such as the communication unit, storage unit, memory unit, etc.

[**0089**] Provide a graphical user interface (GUI) in its host information apparatus **100** to interact with user.

[**0090**] Obtain job object. Job object may be obtained (1) by user input through a GUI, or (2) by using default values stored in a network node or in the output device, or (3) the combination of the above. Default values may be pre-set or may be obtained calculated or generated by the client application as result of communication or negotiation between client application **102**, output device **106** and or server application **112**.

[**0091**] Launch or provide an interface, session or emulation for server application **112**.

[**0092**] Further process output data or print data received from server application **112** before sending the data to output device **106** for final output. The processing may include converting the output data into a form (e.g., format, language, or instruction) more acceptable to or compatible with the associated output device **106**. The processing may also include at least in part one or more raster image processing operations such as rasterization, scaling, color management, color conversion, halftoning, compression, decompression, etc.

[**0093**] Launch, invoke, integrate or involve a helper application to assist at least in part in the processing of the output data.

[**0094**] The above functionalities and process of pervasive output client application **102** are described in further detail in the pervasive output process with reference to **FIG. 4**.

[**0095**] Output device **106** is an electronic system capable of outputting digital content or data content regardless of whether the output medium is a substrate (e.g., paper), display, projection, or sound. A typical example of output device **106** may be a printer, which outputs digital documents containing text, graphics, image or any combination onto a substrate. Output device **106** may also be a display device capable of displaying still images or video, such as, without limitation, televisions, monitors, and projectors. Output device **106** can also be a device capable of outputting sound. Any device capable of playing or reading digital content in audio (e.g., music) or data (e.g., text or document) formats is also a possible output device **106**. A printer (including a fax machine, copier, etc.) is frequently referred to herein as the exemplary output device **106**. However, it should be recognized that the present invention applies also to output device **106** other than printers.

[**0096**] Outputting a data content or output content at an output device (e.g. printers, display devices, projection devices, sound output devices etc.) includes rendering the output content on a specific output medium (e.g., papers, display screens etc). For example, rendering an output content at a printer generates image on a substrate; rendering an output content at a display device generates image on a screen; and rendering an output content at an audio output device generates sound.

[**0097**] A conventional printing system includes three basic components: a raster image processor, a memory

buffer, and a marking engine. The raster image processor converts digital content into a raster suitable for printing; the memory buffer holds the rasterized image ready for printing; and the marking engine transfers colorant to a substrate (e.g., paper).

[**0098**] Marking engine may use any of a variety of different technologies to transform a rasterized image to paper or other media or, in other words, to transfer colorant to a substrate. The different marking or printing technologies that may be used include both impact and non-impact printing. Examples of impact printing may include dot matrix, teletype, daisywheel, etc. Non-impact printing technologies may include inkjet, laser, electrostatic, thermal, dye sublimation, etc.

[**0099**] The marking engine and memory buffer of a printer form its printer engine, which may also include additional circuitry and components, such as firmware, software or chips or chipsets for decoding and signal conversion, etc. Input to a printer engine is usually a final rasterized print data that is generated by the raster image processor. Such input is usually device dependent and printer specific. The printer engine may take this device dependent input and generate output pages.

[**0100**] **FIG. 10A** illustrates an exemplary block diagram of one conventional printing system or printer **1000**A that includes a printer controller **1010**, a memory buffer **1004**, and a marking engine **1006**. The printer controller **1010** includes an interpreter **1001** and a raster image processor **1002**.

[**0101**] The Raster image processor (RIP) may be located within an output device itself (as shown by raster image processor **1002** in **FIG. 10A**) or externally implemented as hardware, software, or a combination (not shown). As an example, RIP may be implemented in a software application or device driver in the information apparatus **100**. A RIP may also reside within a printer controller **1010** (as shown by raster image processor **1002** in **FIG. 10A**), a print server or an output controller **104** of present invention. Examples of raster image processing operations may include image and graphics interpretation, rasterization, scaling, segmentation, color space transformation, image enhancement, color correction, halftoning, compression etc.

[**0102**] When a RIP **1002** is located inside an output device **106**, the RIP **1002** is usually included in a printer controller **1010** (as shown in **FIG. 10A**). A printer controller **1010** may interpret, convert or rasterize input print data in the form of a page description language (e.g., PostScript, PCL), markup language (e.g., XML) or other special document format or language into a final format, language or instructions that printer engine **1008** can understand.

[**0103**] A variety of other page description languages, markup languages, image formats, graphic formats, and file formats may be used as input print data to a printer **1000**A or output device **106**. Examples of possible inputs other than PostScript and PCL may include without limitation, EMF, XML, HTML, among many others. Some printer manufacturers may also employ a combination of proprietary or non-proprietary page description languages, markup languages, file formats, graphics and image formats, color spaces, metafiles, encoding, decoding, compression or decompression etc. for the print data. The print data sent to a printer with printer controller **1010** is usually an intermediate description of a digital document that may require further interpretation, processing or conversion before the print data can be sent to a printer engine **1008** for output. A printer controller **1010** may interpret and process the input intermediate print data into a final format that can be understood by the printer engine **1008**. Regardless of the type of print data, conventionally, a user may need a device-specific driver in his or her information apparatus **100** in order to output the proper language, format, or file that can be accepted by a specific printer or output device **106**

[**0104**] **FIG. 10B** shows another exemplary conventional output system or an output device **1000**B. An output system or output device may include one or more of a printing system or device, a display system or device, a projection system or device, or a sound system or device. In the case that the output system or device is a printer, the printer with reference to **FIG. 10A** does not have a printer controller **1010**. A typical example of printer **1000**B is a lower-cost inkjet printer. RIP operations in this example may be implemented in a software application or device driver in an information apparatus **100** that outputs to such a printer **1000**B. Conventionally, a device-specific driver or application may need to reside in the information apparatus **100** to rasterize and convert the digital document from its original format into final print data (for example a compressed CMKY data with one or more bits per pixel) that can be understood by a particular printer engine **1008**B.

[**0105**] Regardless of type or sophistication level, different output devices **106** conventionally need different printer drivers or output management applications in the information apparatus **100** to provide output capability. Some mobile devices may have limited memory and processing power to store or process multiple device drivers. It may also be infeasible to install or preinstall multiple device dependent or specific printer drivers in such mobile devices. In pervasive output operations of the present invention described below, various device specific drivers or applications may be available and may be executed completely or partially in a remote application server **110**, thereby reducing the workload of information apparatus **100** and realizing device-independent pervasive output.

[**0106**] In one implementation, output device **106** includes at least a communication unit or adapter to interface with information apparatus **100**, as described below in greater detail. Output device **106** may sometimes include more than one communication unit in order to support different interfaces, protocols, or communication standards with different devices. For example, output device **106** may communicate with a first information apparatus **100** through a Bluetooth interface while communicating with a second information apparatus **100** through a parallel interface, and so on. Exemplary hardware components of a wireless communication unit are described below with reference to **FIGS. 2A and 2B**.

[**0107**] Output device **106** may also include an output controller **104** to help manage communication and negotiation processes with information apparatus **100**. Output controller **104** may be dedicated hardware or software or combination of both for at least one output device **106**. Output controller **104** may also be integrated, installed, or

connected externally to one or more output devices **106**. In such cases the output controller **104** may sometimes be referred to as print server or output server.

[**0108**] The difference between output controller **104** and printer controller **1010** should be noted. Printer controller **1010** and output controller **104** are both controllers and are both dedicated hardware and or software for at least one output device **106**. Output controller **104** refers to a controller with feature sets, capabilities, and functionalities of the present invention. Printer controller **1010** may contain functions such as interpreting an input page description language, raster image processing, and queuing, among others.

[**0109**] Output controller **104** may also contain partially or all the features of a printer controller **1010**, plus the feature set, functionalities, capabilities, and processes of present invention.

[**0110**] In one embodiment, output controller **104** does not include a communication unit, but rather utilizes or manages a communication unit residing in the associated output device **106**. In another embodiment, output controller **104** may include or provide a communication unit to output device **106**. For example, an output controller **104** with a wireless communication unit may be installed internally or connected externally to a legacy printer to provide it with wireless communication capability that was previously lacking.

[**0111**] FIGS. **3A-3C** show exemplary implementations of output controller **104** to illustrate that it may be implemented in a variety of ways. The output controller **104** may be connected externally to an output device **106** or integrated internally into the output device **106**. **FIG. 3A** shows that output controller **104** may be implemented as a circuit board or a card that is installed inside an output device **106** and may include software, hardware, or both. **FIG. 3C** shows that output controller **104** may be implemented as an external box or station that is wired or wirelessly connected to an output device **106**. Such an external box or station may contain its own user interface. One example of such an implementation is a print server connected to an output device **106**. **FIG. 3B** shows another configuration in which the functionalities of output controller **104** may be integrated into an existing printer controller **1010** (referred to as "combined controller") which can be internally or externally (not shown) connected to output device **106**. A combined controller **104**B has functionalities of both printer controller **1010** (e.g., input interpretation and or raster image processing) and output controller **104** of the present invention. Under this configuration, the functionalities of output controller **104** and printer controller **1010** may share the same resources, such as processing unit, memory unit, etc.

[**0112**] Other possible implementations of output controller **104** may include, for example, a conventional personal computer (PC), a workstation, and an output server or print server. In these cases, the functionalities of output controller **104** may be implemented using application software installed in a computer (e.g., PC, server, or workstation), with the computer connected with a wired or wireless connection to an output device **106**. Using a PC, server, workstation, or other computer to implement the feature sets of output controller **104** with application software is just another possible embodiment of the output controller **104** and in no way departs from the spirit, scope and process of the present invention.

[**0113**] Regardless of its manner of implementation, the output controller **104** will usually include hardware, software, or a combination. For example, an output controller **104** may include components using one or more or combinations of an application-specific integrated circuit (ASIC), a digital signal processor (DSP), a field programmable gate array (FPGA), firmware, system on a chip, and various communication chip sets. Output controller **104** may also contain embedded processors with software components or embedded application software to implement its feature sets, and functionalities.

[**0114**] Output controller **104** may contain an embedded operating system. With an operating system, some or all functionalities and feature sets of the output controller **104** may be provided by application software managed by the operating system. Additional application software may be installed or upgraded to newer versions in order to, for example, provide additional functionalities or bug fixes.

[**0115**] Output controller **104** typically includes a memory unit, or may share a memory unit with, for example, printer controller **1010**. The memory unit, such as ROM, RAM, flash memory and disk drive among others, may provide persistent or volatile storage. The memory unit may store objects, codes, instructions or data (collectively referred to as software components) that implement the functionalities of the output controller **104**. Part of the software components (e.g., the printer object) may be uploaded to information apparatus **100** during a data output operation.

[**0116**] Functionalities and components of output controller **104** for the purpose of providing pervasive output may include:

> [**0117**] Input components and operations for receiving service requests from a plurality of information apparatuses **100**.
>
> [**0118**] Storage components and operations for storing partial or the entire printer or output object in a memory component.
>
> [**0119**] Transmission components and operations for transmitting partial or the entire printer or output object to the information apparatus **100** requesting pervasive output service. The output controller **104** may transmit the output device object in one or multiple sessions.
>
> [**0120**] Receiving components and operations for receiving output data (e.g., print data) from information apparatus **100** and sending the data to output device **106** or output engine (or printer engine) **1008**.

[**0121**] In addition to the above functionalities, output controller **104** may further optionally include one or more of the following:

> [**0122**] Response components and operations to respond to service request from an information apparatus **100** (e.g., in a discovery process, described below) by providing at least a partial output device object (e.g., printer object for printers).

[0123] Broadcast components and operations to broadcast or advertise the services provided by a host output device **106** to an information apparatus **100** that may request such services.

[0124] Payment components and operations for implementing payment processing and management functions by, for example, calculating and processing payments according to the services requested or rendered to a client (information apparatus **100**).

[0125] Components and operations for implementing job management functionalities such as queuing and spooling among others.

[0126] User interface components and operations for providing a user interface (e.g., display screen, touch button, soft key, etc.) when it is implemented as an external box connected to an output device **106**.

[0127] Security components and operations for implementing security or authentication procedures. For example, the output controller **104** may store in its memory component (or shared memory component) an access control list, which specifies what device or user may obtain service from its host (or connected) output device **106**. Therefore, an authorized information apparatus **100** may gain access after confirming with the control list.

[0128] Processing components and operations to further process the output data. The processing of the output data may include converting the output data into a form (e.g., format, language, and or instruction) at least more acceptable to or compatible with the associated output device **106** or output engine or printer engine **1008**. The processing may also include at least in part one or more raster image processing operations such as rasterization, scaling, color management, color conversion, halftoning, compression, decompression etc.

[0129] When output controller **104** is implemented as firmware, or an embedded application, the configuration and management of the functionalities of output controller **104** may be optionally accomplished by, for example, using controller management software in a host computer. A host computer may be a desktop personal computer (PC), workstation, or server. The host computer may be connected locally or through a network to the output device **106** or the controller **104**. Communication between the host computer and the output controller **104** can be accomplished through wired or wireless communication. The management application software in the host computer can manage the settings, configurations, and feature sets of the output controller **104**. Furthermore, host computer's configuration application may download and or installed application software, software components and or data to the output controller **104** for the purpose of upgrading, updating, and or modifying the features and capabilities of the output controller **104**.

[0130] Output device **106** in one implementation includes or is connected to output controller **106** described above. Therefore, functionalities and feature sets provided by output controller **106** are automatically included in the functionalities of output device **106**. The output device **106** may, however, implement or include other controllers and/or applications that provide at least partially the features and functionalities of the output controller **104**.

[0131] Therefore, the output device **106** may include some or all of the following functionalities:

[0132] Components and operations to receive multiple service requests or queries (e.g., a service request, a data query, an object query etc.) from a plurality of information apparatus **100** and properly respond to them by returning software components, which may contain data, codes, instructions and/or objects.

[0133] Components and operations to receive, from information apparatus **100** device-specific output data containing the digital document intended for output.

[0134] Components and operations to output final output data or print data on a substrate or in another medium such as display screen.

[0135] An output device may further comprise optionally one or more of the following functionalities:

[0136] Components and operations for establishing and managing a communication link with an information apparatus **100** requesting service;

[0137] Components and operations to store and/or further process input print data or output data.

[0138] Components and operations for storing partial or the entire output device object (e.g. printer object) in a memory component.

[0139] Components and operations to advertise or broadcast services provided or available to an information apparatus **100** that may request such services.

[0140] Components and operations for implementing payment processing and management functions by, for example, calculating and processing payments according to the services requested by or rendered to a client (information apparatus **100**).

[0141] Components and operations for implementing job management functionalities such as queuing and spooling among others.

[0142] Components and operations for providing a user interface (e.g., display screen, touch button, soft key, power switch, etc.).

[0143] Components and operations for implementing security or authentication procedures. For example, the output device **106** may store in its memory component (or a shared memory component) an access control list, which specifies what device or user may obtain service from it. Therefore, an authorized information apparatus **100** may gain access after confirming with the control list.

[0144] Components and operations to process the output data. The processing of the output data may include converting the output data into a form (e.g., format, language, and or instruction) more acceptable or compatible to the associated printer controller **1010**, output engine or printer engine **1008**. The

processing may also include at least partially one or more raster image processing operations.

[0145] Application server **110** is a node on network **108**. An application server **110** may include computing capability, data storage capability, and mechanisms for servicing requests from a plurality of client computers (referred to as clients), including the information apparatus **100**, needing computational or data storage resources. A server typically includes processing unit, memory unit, storage unit, input/output control unit, and a communication unit, among others. Application server **110** may also include an interface to interact with users. The interface may be implemented with, for example, display screen, touch-sensitive screen, keyboard, mouse, stylus, push button, microphones and speakers among others.

[0146] Application server **110** preferably includes at least an operating system for supporting a plurality of application software to implement the functionalities of the application server **110** and to provide services to its clients (e.g., information apparatus **100**). To provide services to multiple clients at the same time, the application server **110** may run a multi-user operating system that enables multiple concurrent users to log on and run applications in separate, protected sessions.

[0147] Application server **110** typically includes a server application **112** that may include software and data to be executed in the processing unit of a server node. The server application **112** may include one or more of the following functionalities:

[0148] Components and operations to receive data and/or objects (with at least a output device object and a document object) from client application **102**.

[0149] Components and operations to process the objects received to generate device-dependent output data acceptable to one or more output devices **106** selected by a user. The server application **112** may perform the processing function independently or in combination with other applications.

[0150] Components and operations to transmit output data back to the client application **102**.

[0151] The server application **112** may optionally include one or more of the following functionalities:

[0152] Components and operations to obtain digital document (output content) from other network node if it is not included or completely included in the document object received from client application **102**.

[0153] Components and operations to obtain output device parameters if they are not included or completely included in the output device object received from client application **102**.

[0154] Components and operations to obtain job object or relating parameters if it is not included or completely include in the composite message received from client application **102**. The server application **112** may obtain job object by querying user or by assuming default values. For example, if a user did not specify print or output range, the server application **112** may assume the default by printing or outputting all pages.

[0155] Components and operations to manage and coordinate communication with an information apparatus **100** requesting output service.

[0156] Components and operations to identify, invoke and or incorporate appropriate applications (residing in the application server **112** or in other network nodes) to help interpret and or process the digital document obtained and generate output data. The processing of the digital document may include converting the digital document into an output data related at least in part to the output device object. The processing may also include at least in part one or more raster image processing operations such as rasterization, scaling, color management, color conversion, halftoning, compression, decompression, etc.

[0157] Components and operations to interact with (e.g., store, update, verify, etc.) one or more databases that store profiles of information apparatuses **100** and/or users who have subscribed to output service provided by the server application **112**.

[0158] Components and operations to implement an independent computing architecture where the execution of the server application **112** may occur entirely in the application server **110**. The GUI of the application **112**, keystrokes and mouse clicks are transmitted over the network **108** to and from the client such as a user's information apparatus **100**.

[0159] It should be noted that, in the example shown here, server application **112** is illustrated as the only application in application server **110**. But in actuality, more than one application may exist in the application server **110** and the applications may provide various services to different clients. It is also possible that the functionalities and feature sets of the server application **112** may be implemented with multiple applications residing in the same server, multiple servers (e.g., in a server farm), or network nodes or combination.

[0160] Content server **114** may represent one of a plurality of server nodes on network **108** that may store digital documents **116**. The digital documents stored in content server **114** may be viewed or edited by a user using an information apparatus **100**. As an example, the content server **114** may be a web server that hosts a plurality of web pages written in mark up languages such as HTML, WML, XML, HDML, CHTML, among others. A user may view web pages using an Internet browsing application such as Internet Explorer or Netscape Navigator, a WAP browser, etc. As another example, the content server **114** may be a file server that allows multiple clients to store and share digital files with appropriate security or authentication procedures. These digital files or documents may contain one or more of image, text, graphics, sound and video. The files may be saved in various file formats (e.g., MS Word, Excel, PowerPoint, PDF, Postscript, JPEG, GIF, MPEG, etc.). A user may need to have appropriate application on his/her information apparatus **100** to access, view and edit these files.

[0161] It should be noted that in **FIG. 1** content server **114** and application server **110** are shown as distinct server nodes. However, it is possible that application server **110** and content server **114** belong to the same domain, or the

functionalities of these two servers may be implemented with different applications executed in a single network node or server.

[0162] Further description of the functionalities and feature sets of different devices and applications illustrated in **FIG. 1** is provided below in reference to a pervasive output process shown in **FIG. 4**.

[0163] **FIGS. 2A and 2B** are block diagrams illustrating two exemplary configurations of hardware components of wireless communication units.

[0164] Referring to **FIG. 2A, a** radio adapter **200** may be implemented to enable data/voice transmission among devices (e.g., information apparatus **100** and output device **106**) through radio links. A RF transceiver **214** coupled with antenna **216** is used to receive and transmit radio frequency signals. The RF transceiver **214** also converts radio signals into and from electronic signals. The RF transceiver **214** is connected to a RF link controller **210** by an interface **212**. The interface **212** may perform functions such as analog-to-digital conversion, digital-to-analog conversion, modulation, demodulation, compression, decompression, encoding, decoding, and other data or format conversation functions.

[0165] RF link controller **210** implements real-time lower layer (e.g., physical layer) protocol processing that enables the hosts (e.g., information apparatus **100**, output controller **104**, output device **106**, etc.) to communicate over a radio link. Functions performed by the link controller **210** may include, without limitation, error detection/correction, power control, data packet processing, data encryption/decryption and other data processing functions.

[0166] A variety of radio links may be utilized. A group of competing technologies operating in the 2.4 GHz unlicensed frequency band is of particular interest. This group currently includes Bluetooth, Home radio frequency (Home RF) and implementations based on IEEE 802.11 standard. Each of these technologies has a different set of protocols and they all provide solutions for wireless local area networks (LANs). Interference among these technologies could limit deployment of these protocols simultaneously. It is anticipated that new local area wireless technologies may emerge or that the existing ones may converge. Nevertheless, all theses existing and future wireless technologies may be implemented in the present invention without limitation, and therefore, in no way depart from the scope of present invention.

[0167] Among the current available wireless technologies, Bluetooth may require relatively lower power consumption. Bluetooth has its own protocol stack and is designed for short range (10 meters), point-to-multipoint voice and data transfer. It is based on a frequency-hopping version of spread spectrum. Seventy-nine hop frequencies are utilized beginning at the lowest frequency of 2402 MHz and each of the 79 hop frequencies is 1 MHz above the next lower frequency. Bluetooth-enabled devices operate in piconets, in which several devices, using the same hopping pattern or sequence, are connected in a point-to-multipoint system (piconet). There is one device (master) in each piconet that determines how the bandwidth is allocated to other devices (slaves). As many as 10 piconets of 8 devices each can operate simultaneously.

[0168] Referring to **FIG. 2**B, one or more infrared (IR) adapters **220** may be implemented to enable data transmission among devices through infrared transmission. The IR adapters **220** may be conveniently implemented in accordance with the Infrared Data Association (IrDA) standards and specifications. In general, the IrDA standard is used to provide wireless connectivity technologies for devices that would normally use cables for connection. The IrDA standard is a point-to-point (vs. point-to-multipoint as in Bluetooth), narrow angle, ad-hoc data transmission standard designed to operate over a distance of 0 to 1 meter and at speeds up to 4 Mbps.

[0169] Configuration of infrared adapters **220** may vary depending on the intended rate of data transfer. **FIG. 2B** illustrates one embodiment of infrared adapter **220**. Transceiver **226** receives/emits IR signals and converts IR signals to/from electrical signals. A UART (universal asynchronous receiver/transmitter) **222** performs the function of serialization/deserialization, converting serial data stream to/from data bytes. The UART **222** is connected to the IR transceiver **226** by encoder/decoder (ENDEC) **224**. This configuration is generally suitable for transferring data at relatively low rate, for example 115.2 kbps or below. Other components (e.g., packet framer, phase-locked loop) may be needed for higher data transfer rates.

[0170] **FIGS. 2A and 2B** illustrate exemplary hardware configurations of wireless communication units. Such hardware components may be included in devices (e.g., information apparatus **100**, output controller **104**, output device **106**, etc.) to support various wireless communications standards. Wired links, however, such as parallel interface, USB, Firewire interface, Ethernet and token ring networks may also be implemented in the present invention by using appropriate adapters and configurations.

[0171] FIGS. **3**A-**3**C illustrate various configurations and implementations of output controller **104**.

[0172] In the configuration illustrated by **FIG. 3**A, output controller **104**A may be installed, for example as one or more boards or cards containing both hardware and software, inside output device **106**(A). The output device **106**(A) may or may not have inside it a printer controller **1010** (**FIG. 10**). In the case that output device **106**(A) includes a printer controller **1010** (not shown), the output controller **104** may co-exist with the printer controller **1010** and other components of the output device **106**(A). One example of this implementation is to connect or cascade output controller **104**A sequentially or serially with the printer controller **1010**.

[0173] In another implementation shown by **FIG. 3**B, the functionalities of output controller **104** and printer controller **1010** (**FIG. 10**) may be combined into a single controller, sometimes referred to as combined controller **104**(B). The combined controller **104**(B) may be implemented, for example, as one or more boards or cards containing both hardware and software, inside output device **106**(B). This implementation may help to reduce the cost of material when compared to implementing two separate controllers. As an example, the combined controller **104**(B) may share common processors, memories, and storage units to run the applications and functionalities of the two types of controllers and, therefore, may have lower component cost. The combined controller can also be implemented as an external box or station (not showed) connected to the output device **106**. This connection could be wired or wireless.

[0174] In the third implementation shown in **FIG. 3**C, the output controller **104**C may be implemented in a separate box or server or station connected externally to output device **106**C. The communication link between output controller **104**C and output device **106**C may be a wired or wireless link. The output device **106**C may or may not have inside it a printer controller **1010** (**FIG. 10**). If the output device **106**C does not contain a printer controller **1010** inside, the externally connected output controller **104**C may optionally contain the functionalities of printer controller **1010** and therefore, provide raster image processing capability as an additional feature.

[0175] The above are exemplary implementations and configurations of output controller **104**. Other implementations are also possible. For example, partial functionalities of output controller **104** may be implemented in an external box or station while the remaining functionalities may reside inside an output device **106** as a separate board or integrated with a printer controller **1010**. As another example, the functionalities of output controller **104** may be implemented into a plurality of external boxes or stations connected to the same output device **106**. As a further example, the same output controller **104** may be connected to service a plurality of output devices **106**. Variously implemented output controllers **104** should not depart from the spirit and scope of the present invention provided that they all support the functionalities and feature sets described herein.

[0176] **FIG. 4** is a flow diagram of a pervasive output process **401** of the present invention. Pervasive output process **401** allows an information apparatus **100** to output digital content or document in its original form to an output device **106** regardless of processing power, display screen size, or memory space of information apparatus **100**. Pervasive output process **401** may include or utilize:

[0177] A client application **102**, in an information apparatus **100**, obtaining objects; with at least one object including a document object and another object including an output device object (or printer object in the case of a printer);

[0178] A client application **102** transmitting objects to a server application **112**;

[0179] A server application **112** obtaining and processing the document object and converting it into output data, reflecting at least in part a relationship to said output device object;

[0180] A server application **112** transmitting output data to the information apparatus **100** including a client application **102**;

[0181] The information apparatus **100** including a client application **102** transmitting output data to an output device **106**;

[0182] An output device **106** generating output with the output data.

[0183] Pervasive output process **401** may be initiated by a user in step **400**. Typically, a user initiates output process **401** by invoking a client application **102** in his/her information apparatus **100**. The client application **102** may be launched as an independent application or it may be launched from other applications **103** (such as from a document browsing, creating or editing application) or as part of or component of or a feature of another application **103** residing in the same information apparatus **100**. When launched from another application **103**, the client application **102** may obtain information, such as a partial or an entire document object, from that another application **103**. This can be accomplished, for example, by one or combinations of messages or facilitated through an operating system or a particular object or component model etc. The client application maybe a device driver in this example.

[0184] During output process **401**, a user may need to select one or more output devices **106** for output service. An optional discovery process step **404** may be implemented to help the user select an output device **106**.

[0185] During the discovery process step **404**, a user's information apparatus **100** may (1) search for available output devices **106**; (2) provide the user with a list of available output devices **106**; and (3) provide means for the user to choose one or more output devices **106** to take the output job. An exemplary discovery process **404** is described below in greater detail with reference to **FIG. 5**.

[0186] The discovery process **404** may sometimes be unnecessary. For example, a user may skip the discovery process **404** if he or she already knows the output device (e.g., printer) **106** to which the output is to be directed. In this case, the user may simply connect the information apparatus **100** to that output device **106** by wired connections or directly point to that output device **106** in a close proximity such as in the case of infrared connectivity. As another example, a user may pre-select or set the output device or devices **106** that are used frequently as preferred defaults. As a result, the discovery process **404** may be partially or completely skipped if the default output device **106** or printer is found to be available.

[0187] In stage **406**, the client application **102** may interact with output device **106**, the user, and or other applications residing in the same information apparatus **100**, to obtain (1) a document object, (2) an output device object and (3) any other optional objects such as a job object. These objects may be obtained sequentially, concurrently, or in any order. It should be noted that some of the objects or partial objects might have been acquired in prior steps. For example, the client application **102** may have obtained partially or entirely document objects or related information when the client application **102** was launched from or by another application. As another example, a partial output device object may have been uploaded to the information apparatus **100** during the optional discovery process **404**. The client application **102** may create a composite message including these objects (document object, output device object and other optional objects) and transmit the composite message to server application **112** for processing, as described below in greater detail with reference to **FIG. 6**.

[0188] The server application **112**, after receiving such a composite message from the client application **102**, may in step **408** processes the document object or objects contained in the composite message and convert it or them into output data. Additional helper applications may be needed to help in this processing task. The processing and generation of output data may reflect at least in part a relationship to the output device object and or job object contained in the composite message received from client application **102**. The output data generated may be transmitted back to the

information apparatus **100**, requesting output service or process **401** via network **108**. An exemplary implementation of a server application process is described below with reference to **FIG. 7**.

[**0189**] In step **410**, information apparatus **100** transmits output data, with or without further processing, to the selected output device **106** through a local communication link **116**. Preferably, the output data is in a format or language acceptable to or compatible with the output device **106** selected by the user. When receiving output data from information apparatus **100**, an output device **106** may simply buffer the output data before sending it to output engine (or printer engine) **1008** for final output. In cases where an output device **106** includes or is connected to a printer controller **1010**, output controller **104** or a combined controller, such controllers may further process the output data received from information apparatus **100** before sending the output data to printer engine **1008** for final output. Further processing, if it exists, may include at least partially conversion operations and or raster image processing operations on the output data. The output controller **104** may also provide queuing or spooling or other job management features. An exemplary implementation of such an output process is described below with reference to **FIG. 8**.

[**0190**] The steps included in pervasive output process **401** may proceed automatically when a user requests output service. Alternatively, a user may be provided with options to proceed, cancel, or input information at each and every step. For example, after the client application process step **406** is finished, the user may be asked whether or not he or she wishes to continue to communicate with a remote application server. As another example, a user may cancel the output service at any time by, for example, indicating a cancellation signal or command or by terminating the client application or by shutting down the information apparatus **100** etc.

[**0191**] A printer is used as a primary output device **106** in this example. It should be recognized, however, that pervasive output process **401** described above can be similarly applied to other output devices such as fax machines, digital copiers, display screens, monitors, televisions, projectors, voice output devices, among others.

[**0192**] **FIG. 5** is a flow diagram of an exemplary implementation of a discovery process **520**, which may be an optional step to help a user locate one or more output devices **106** for an output job. The discovery process **520** may, however, be skipped partially or entirely. Implementation of discovery process **520** may require compatible hardware and software components residing in both the information apparatus **100** and the output device **106**.

[**0193**] The discovery process **520** may include or utilize:

[**0194**] A client application **102** in an information apparatus **100** communicating with available output devices **106** to obtain information such as partial or complete output device object or objects.

[**0195**] A client application **102** that provides to the user information on each available and or compatible output device **106**.

[**0196**] User selection or determination by a client application **102** (automatically or not) of one or more devices **106** for output service from the available or compatible output devices **106**.

[**0197**] Various protocols and or standards may be used during discovery process **520**. Wireless communication protocols are preferred. Wired communication, on the other hand, may also be implemented. Examples of applicable protocols or standards may include, without limitation, Bluetooth, HAVi, Jini, Salutation, Service Location Protocol, and Universal Plug-and-play among others. Other proprietary protocols or combination may also be implemented in the discovery process **520**. However, these different protocols, standards, or combination shall not depart from the spirit and scope of present invention.

[**0198**] In one implementation an application (referred here for simplicity of discussion as a "communication manager," not shown) residing in the information apparatus **100** helps communicate with output device **106**, manage service requests and discovery process **520**. The communication manager may be a part of or a feature of the client application **102**. Alternatively or in combination, the communication manager may also be a separate application. When the communication manager is a separate application, the client application **102** may have the ability to communicate, manage or access functionalities of the communication manager.

[**0199**] The discovery process **520** may be initiated manually by a user or automatically by a communication manager when the user requests an output service with information apparatus **100**.

[**0200**] In the optional step **500**, a user may specify searching or matching criteria. For example, a user may indicate to search for color printers and or printers that provide free service. The user may manually specify such criteria each time for the discovery process **520**. Alternatively or in combination, a user may set default preferences that can be applied to a plurality of discovery processes **520**. Sometimes, however, no searching criteria are required—the information apparatus **100** may simply search for all available output devices **106** that can provide output service.

[**0201**] In step **501**, information apparatus **100** searches for available output devices **106**. The searching process may be implemented by, for example, an information apparatus **100** multi-casting or broadcasting or advertising its service requests and waiting for available output devices **106** to respond. Alternatively or in combination, an information apparatus **100** may "listen to" service broadcasts from one or more output devices **106** and then identify the one or more output devices **106** that are needed or acceptable. It is also possible that multiple output devices **106** of the same network (e.g., LAN) register their services with a control point (not shown). A control point is a computing system (e.g., a server) that maintains records on all service devices within the same network. An information apparatus **100** may contact the control point and search or query for the needed services.

[**0202**] In step **502**, if no available output device **106** is found, the communication manager may provide the user with alternatives **504**. Such alternatives may include, for example, aborting the discovery process **520**, trying discovery process **520** again, temporarily halting the discovery process **520**, or being notified when available output device

106 is found. As an example, the discovery process **520** may not detect any available output device **106** in the current wired/wireless network. The specified searching criteria (if any) is then saved or registered in the communication manager. When the user enters a new network having available output devices **106**, or when new compatible output devices **106** are added to the current network, or when an output device **106** becomes available for any reason, the communication manager may notify the user of such availability.

[**0203**] In step **506**, if available output devices **106** are discovered, the communication manager may obtain some basic information, or part of or the entire output device object, from each discovered output device **106**. Examples of such information may include, but not limited to, device identity, service charge, subscription, service feature, device capability, operating instructions, etc. Such information is preferably provided to the user through the user interface (e.g., display screen, speaker, etc.) of the information apparatus **100**.

[**0204**] In step **508**, the user may select one or more output devices **106** to take the output job. If the user is not satisfied with any of the available output device **106**, the user may decline the service. In this case, the user may choose to try again in step **510** with some changes made to the searching criteria. Alternatively, the user may choose to terminate the service request.

[**0205**] In step **512**, with one or more output devices **106** selected or determined, the communication link between information apparatus **100** and the selected output device or devices **106** may be "locked". Other output devices **106** that are not selected may be dropped. The output process **520** may then proceed to a client application process **601** referenced in step **406** of **FIG. 4**.

[**0206**] **FIG. 6** is a flow diagram of an exemplary client application process **601** that may include or utilize:

[**0207**] A client application **102** that obtains an output object or objects as a result of communication and or negotiation with an output device **106**.

[**0208**] A client application **102** that obtains a document object (1) from user input or selection, or (2) from other applications (e.g., document browsing application) residing in the same information apparatus **100**.

[**0209**] A client application **102** that coordinates with server application **112** to manage the process of communication and transmission of a composite message (including at least a printer object and a document object) to the server application **112** for further processing.

[**0210**] Step **600** indicates that client application **102** obtains a document object. To assemble or create a document object, the client application **102** may need to obtain output content (or a digital document) or pointers to the output content, or both. In cases where the output content is downloaded or partially downloaded to a user's information apparatus **100**, the client application **102** may include both the output content and pointers to the output content in its document object. In cases where the output content is not downloaded to the information apparatus **100**, the client application **102** may include only pointers to the output content in the document object.

[**0211**] There are various ways that a client application **102** may obtain a digital document or a pointer or reference to the digital document. In one embodiment, the client application **102** may provide a GUI with which a user can directly input the pointer or reference (e.g., URL, IP address, filename, path, etc.) of a digital document stored locally or in a network node. This manual process may be facilitated by, for example, providing a GUI with which users may select one or more pointers from a list of pointers or references of digital documents stored locally or in a remote network node. For instance, through a GUI provided by the client application **102**, a user may see and select pointers of digital documents stored in a remote file server. In this case, the document object created by the client application **102** may contain only pointers or references to the output content.

[**0212**] In another embodiment, the client application **102** may obtain output content or pointer to output content from another application in the same information apparatus **100**. As an example, a user may (1) launch the client application **102**, and (2) invoke another application **103** (e.g., document editing and or browsing application) residing in the same information apparatus **100** to view or download the digital document. As another example, a user may (1) run another application **103** (e.g., document editing and or browsing application) residing in the same information apparatus **100** to view or download the digital document; and (2) launch or invoke the client application **102**. In these cases, the client application **102** may communicate with another application **103** (e.g., document browsing application) to obtain pointers to the digital document and or the digital document itself (if it has been downloaded locally for viewing) to be included in document object.

[**0213**] It should be noted that the document object may have been partially or entirely obtained by the client application **102** in previous steps such as in step **400** of **FIG. 4** in which the client application **102** is initiated. As an example, a user may (1) view or download a digital document (stored locally or in a network node) by using a document browsing application on the information apparatus **100**, and (2) request output service by launching the client application **102** (as in step **400**). The client application **102** may then communicate with the document browsing application to obtain the document object (including digital document and or pointers to the digital document). In this case, step **600** may be partially or entirely skipped.

[**0214**] In some instance such as in document browsing applications, a user may be limited to viewing or downloading only part of, or a reduced version of, the original digital document stored in a network node. This may be due to small screen size, limited bandwidth, memory size, and lack of application support, among other issues. In these cases, the client application **102** may allow the user to choose to output the original digital document or the reduced sized document. In the case of the original document, the document object created by the client application **102** may, for example, contain only references or pointers to the original output content stored in a server or network node. In the case where user wants to output the reduced version document that has been downloaded locally, the client application **102** may include in the document object one or more of (1) the

obtained reduced output content (2) the reference to the reduced content in the server or network node, and (3) pointer or pointers to the original output content.

[0215] In another instance, for example, with a document browsing application, a user may see only the name or path to output content without being able to download or open it due to, for example, no compatible applications residing on the information apparatus **100**, small display screen etc. According to the present invention, however, a user may still be able to output the original digital document by indicating to the client application **102** the pointer or reference of the output content. For example, the user may select or highlight the name or path of the output content through a GUI provided by the document browsing application. In this case, the document object created by the client application **102** may contain only pointers to the output content.

[0216] A document object may also contain instructions. Instructions in an object may provide description, operation, and status information of the content or data of the document object, etc. For example, instructions may provide information about the changes or differences between an output content included in a document object relative to the original output content (stored in a network node) that is pointed to by the reference field in the document object. The output content included in a document object may be an edited or altered version while the output content referred to by the pointers may be the original version.

[0217] It should be noted that some output content may be publicly available to all users (e.g., generic web pages) while other output content may be restricted to one or a group of users (e.g., secure documents stored in a corporate network). In the latter case, authentication information such as a password, user name, id number, biometric information, digital certificate or security key, among others, may need to be provided to the server application **112** for accessing and or fetching the digital document with reference to a pointer or reference. In one example, such authentication information may be included in document object, printer object, or job object. In another example, server application **112** may prompt the user to enter authentication information through a UGI in information apparatus **100** when necessary.

[0218] Step **602** indicates whether an output device object is obtained. In the case where an output device object is not obtained, then in step **604**, with one or more output devices **106** selected, the client application **102** may communicate with the selected output device or devices **106** to upload output device objects or related information stored in memory or storage components of the output device or devices **106**. It should be noted that a partial or entire output device object or related information might have been already obtained by the client application **102** during the prior optional discovery process (**520** and step **402** in **FIG. 4**). In this case, step **604** may be partially or entirely skipped.

[0219] To successfully obtain the output device object or objects from the selected output device or devices **106**, several additional optional processes may be involved. As an example, authentication may be necessary when the selected output device **106** provides service to a restricted group of users. A simple authentication may be implemented by, for example, comparing the identity for the information apparatus **100** with an approved control list of identities or elements stored in the output device **106**. Other more complex authentication and encryption schemes may also be used. Information such as user name, password, ID number, signatures, security keys (physical or digital), biometrics, fingerprints, voice, among others, may be used separately or in combination as authentication means. Such identification and or authentication information may be manually provided by the user or automatically detected by the selected output device or devices **106**. With successful authentication, a user may gain access to all or part of the services provided by an output device **106**. The output device object that the client application **102** obtains may vary according to the type or quality of service requested or determined. If authentication fails, it is possible that a user may be refused partially or completely all access to the service. In this case, the user may be provided with alternatives such as selecting another output device **106** or alternative services.

[0220] Another optional process is that a user may be asked to provide payment or deposit or escrow before, during or after output service. Examples of payment or deposit may include cash, credit card, bankcard, charge card, smart card, electronic cash, among others. The output controller **104** may provide payment calculation or transaction processing as optional feature sets.

[0221] It should be noted that a plurality of information apparatuses **100** may request to obtain output device object or objects from the same output device **106** at the same time or at least during overlapping periods. The output device **106** may have components or systems to manage multiple communication links and provide the output device object or objects concurrently or in an alternating manner to multiple information apparatuses **100**. Alternatively, an output device **106** may provide components or systems to queue the requests from different information apparatuses **100** and serve them in a sequential fashion according to a scheme such as first come first serve, quality of service, etc. Multi-user communication and service management capability with or without queuing or spooling functions may be implemented by, for example, the output controller **104** as optional feature sets.

[0222] Step **606** indicates that the client application **102** may optionally obtain a job object. A job object may include a user's preferences and parameters relating to the output job or process. The client application **102** may obtain a job object by, for example, capturing a user's output preferences through a GUI. In one instance, the client application **102** may provide a universal GUI to a user regardless of what output device **106** is selected. Through such an interface, the user may specify some device-independent parameters such as page range, number of cards per page, number of copies, etc. Alternatively or in combination, the client application **102** may incorporate device-dependent features and preferences into the GUI provided to user. The device dependent portion of the GUI may be supported partially or entirely by some information contained in or provided by the output device object obtained from the selected output device **106**. Examples of such device dependent features may include quality of service, service fee, print quality, color or gray-scale, duplex or single sided, output page size, among others.

[0223] It is possible that some or all components, attributes or fields of a job object have default values. The client application **102** may have certain defaults such as

those hard-coded in software or hardware or pre-configured by the user or a manufacturer. In addition, the client application **102** may access a file to obtain default values or by other means such as communicating with the output device **106**, the server application **112** or other applications in the information apparatus **100** or in a network node. The client application **102** may also create or calculate certain default values based on the information it has or obtained during the output process. These default values may be related, at least in part, to the output device object and or the server application **112** involved in the output process. In some instances, a user may or may not have an opportunity to change or overwrite some or all defaults. The client application **102** may obtain and use some or all defaults with or without user intervention or knowledge.

[**0224**] In step **608**, the client application **102** may create or assemble a composite message. A composite message may be any type of data transferred across network **108** that may include one or more transmissions. A composite message typically includes partially or entirely the objects (with some default values) obtained by the client application **102** in previous steps.

[**0225**] In step **610**, the client application **102** transmits the composite message to server application **112**. The client application **102** may communicate with the server application **112** using one or more or a combination of standard network protocols such as WAP, Web Clipping, I-Mode, TCP/IP, SPX/IPX, PPP, NetBEUI, Apple Talk, among others. Proprietary network protocols or a combination that includes them may also be used. The communication link between information apparatus **100** and application server **112** may be implemented with one or a combination of standard network connections and communication links such as telephone lines, LAN or WAN links (T1, T3, 56 kb, X.25, etc.), broadband connections (ISDN, Frame Relay, ATM etc), wireless connection (radio link, inferred, microwave, etc.) as well as the Internet or corporate Intranets.

[**0226**] In addition to the composite message, the client application **102** and server application **112** may also exchange various types of messages back and forth to, for example, request service, confirm service availability, configure protocol stack, or confirm or acknowledge receipt of the previous message, among other messages. The server application **112** may also prompt the client application **102** to send or resend some components or objects of the composite message if such components or objects are incomplete, missing or corrupted. The server application **112**, however, may also try to fill in default values (if available) for some or all of the information missing in the composite message. Encode/decode, compression/decompression and/or encryption techniques may be used to facilitate the transmission of the composite message.

[**0227**] There is a possibility that the application server **112** only provides services to users who have subscribed to such services. In this case, a user may be prompted to log in or sign up when he or she requests service from the server application **112**.

[**0228**] The server application **112** may receive composite messages or service requests from a plurality of client applications **102** at the same time or at least during overlapping periods. It is therefore, beneficial that the server application **112** runs on a multi-user operating system that enables multiple concurrent users to run applications in separate, protected sessions.

[**0229**] It should be recognized that **FIG. 6** illustrates only one example of the client application process **406** described with reference to **FIG. 4**. Other implementations are also possible. As an example, the client application **102** may obtain document objects, output device objects and optional job objects in any combination of order or sequence, or may obtain them concurrently. As another example, step **610** (transmitting composite message) may proceed before step **608** (creating composite message) is finished. In other words, during the process when client application **102** is obtaining objects from the user, output device **106**, or other applications in the same information apparatus **100**, the client application **102** may also communicate with the server application **112** concurrently or in alternating times to transmit partially or completely the objects it has obtained.

[**0230**] **FIG. 7** is a flow diagram of an exemplary server application process **701**. Typically, an application server **110** is a much more powerful computing device than a mobile information apparatus **100**. The server **110** may contain a plurality of applications to interpret, process, or rasterize digital documents into output data. Therefore, in one implementation the core computation of an output process is executed in the application server **110**. The server application process **701** may include or utilize:

[**0231**] A server application **112** that receives a composite message (including an output device object and a document object) from client application **102**.

[**0232**] A server application **112** that processes the objects received to generate device-dependent output data acceptable to one or more output devices **106** selected by a user. The server application **112** may perform the processing function independently or in combination with other applications.

[**0233**] A server application **112** that transmits output data back to the information apparatus **100** requesting output service or services.

[**0234**] In step **700**, server application **112** receives a composite message from client application **102**. As mentioned earlier, the composite message may include one or more of a document object, an output device object and an optional job object.

[**0235**] After receiving the composite message from client application **102**, the server application **112** may process the document object and convert it into output data. The output data generated is preferred to be in a format or language acceptable or compatible to the output device or devices **106** selected by the user.

[**0236**] In some cases, the document object or output device object in a composite message may include only references, pointers and or instructions. Such a configuration may, for example, reduce the size of the composite message and therefore make it easier to transmit through a network. In another case, the document object or output device object in a composite message may include missing or incomplete information. Digital document (e.g., output content) or output device parameters may not be available or completely available in the document object or the output device object,

respectively. In these cases, as shown in step **704**, the server application **112** may need to find or fetch partially or entirely the output content or device parameters from network nodes or otherwise obtain the content or parameters according to the pointers, references or instructions provided by the objects. For example, an output device object received in a composite message may contain insufficient output device parameters only the brand name and model number of a selected output device **106** may be provided. In this case, the server application **112** may either find or fetch the missing output device parameters according to pointers or references provided in the output device object, or the server application **112** may consult a translation service or application in order to obtain the complete output device parameters based on the partial or limited device information that is available. If no further information or data can be found or obtained, predefined defaults may be used to fill in missing information about the output device **106** or alternatively a warning or an error message may be provided to the user for either continuation or cancellation of the process.

[**0237**] Step **704** (obtaining output content and or device parameters) may be entirely or partially skipped if the output content and output device parameters are already included in the document object and output device object, respectively.

[**0238**] In order to process a digital document into device dependent output data, step **706** shows that the server application **112** may need to involve one or more helper applications. A helper application is any application in the server **110** or other network node that participates, helps or assists in the output process of the present invention. Examples of helper applications may include, software components, software applications, device drivers, printer drivers, etc. The helper application or applications may perform one or more of the following tasks among others:

[**0239**] Open, parse, or interpret a particular digital document format or language.

[**0240**] Convert a digital document into an intermediate format, language or data.

[**0241**] Process an input digital document into a raster format including, as an example, one or more raster image processing operations such rasterization, scaling, color correction, color matching, segmentation, halftoning, compression, decompression etc.

[**0242**] Convert or encode rasterized data into a device specific output data.

[**0243**] The helper applications may participate in the output process in various ways. As an example, a server application **112** may involve one or more helper applications to decode, parse, interpret, and or process a digital document into an intermediate file, format, language or data. And then the server application **112** may involve one or more helper applications to further convert the intermediate result into output data. In another example, the server application **112** may involve two or more groups of helper applications. The first group of helper applications may decode, interpret, and process a digital document into intermediate file, format, language or data. The second group of helper applications may convert the intermediate results into output data. A group may consist of one or more applications.

[**0244**] The server application **112** may identify, locate, invoke, launch, and or integrate the appropriate helper applications or its components based on the information (e.g., output device parameters) provided in an output device object. A translation service (e.g., database application, directory service) may provide information to the server application **112** as to where to find, obtain, or use the helper applications. The helper applications may reside locally in the application server **110** or remotely in another network node. If a helper application is located in another network node, the server application **112** may acquire or download the helper application to the application server **110**. Alternatively, the server application **112**, considering application availability or load balancing, may direct the digital document to be processed in another server node where the helper application or applications are run.

[**0245**] In some cases, however, the server application **112** may have the ability to process and generate the correct output data without involving additional helper applications and step **706** may be skipped. As an example, the server application **112** may already contain software components and feature sets capable of supporting and generating, different print data or output data formats or language such as PostScript or PCL or XML etc. Therefore, if the output device object indicates that the output device **106** is a PostScript printer or any other that it supports, the server application **112** may not need to invoke, or download or incorporate a helper application or a helper application component such as a PostScript driver. The server application **112** may just proceed and convert the digital document into a PostScript file as print data. PostScript is used here as an example, other formats, languages, data used for other types of print data or output data follows the same fashion.

[**0246**] When fetching output content or device parameters or software components or drivers from other network nodes, the server application **112** may need authentication information in order to gain access. Such authentication information may be already stored in the server application **112** (e.g., in a user profile), in the received objects (e.g., job object, output device object or document object), or the user may input authentication information as needed.

[**0247**] While processing digital content in step **708**, the server application **112** may also distribute or send a presentation or GUI to a client device (e.g., information apparatus **100**) to inform the user of the processing status (shown in step **709**). The presentation or GUI in the client device may also capture user's preferences and inputs such as login information, security information among other preferences or inputs. This client/server implementation described here may provide an efficient computing environment. For example, logic for the server application **112** may run in the application server **110** and its distributed user interface may run in the client device (e.g., information apparatus **100**). Therefore, only data relating to keystrokes, mouse clicks and screen updates may, as an example, travels through the network. This client/server implementation may reduce bandwidth requirements. Such a GUI may be implemented by, for example, instructing the client application **102** to launch on the display screen of the client information apparatus **100** a plurality of windows emulating the user interface of the server application **112**. The presentation of the server application interface may be displayed efficiently through these windows in the client application **102**.

[**0248**] If the server application **112** receives no job object or an incomplete job object in a composite message, the

server application **112** may assume default values for the job object or optionally, as shown in step **709**, it may also launch a GUI in the information apparatus **100** as described above to obtain partially or entirely the job object fields (job preference) from the user. Information provided in the job object may be used in the process of generating output data (step **708**).

[**0249**] It is noted that the output data (or print data in the case of a printer) generated by the server application **112** may or may not be the final output data or the final print data that can feed directly into an output engine or a printer engine **1008**. The output data sent to output device **106** through information apparatus **100** may in some cases need further processing by an application in the information apparatus **100** or by the output controller **104** or printer controller **1010** or combination before finally being sent to output engine or printer engine **1008**. All these different combinations of processes and tasks or load distributions are possible implementations that shall fall within the scope and spirit of the present invention.

[**0250**] In step **710**, the server application **112** transmits output data to the information apparatus **100** through network **108**. Encryption techniques may be applied to the output data to ensure security. Compression/decompression, encode/decode may also be implemented to facilitate the transmission of output data over network **108**.

[**0251**] **FIG. 8** is a flow diagram of an exemplary final output process **801** for pervasive output, which may include or utilize:

[**0252**] A server application **112** transmits output data to information apparatus **100**.

[**0253**] An information apparatus **100** transmits output data, with or without further processing, to the output device **106** selected by the user.

[**0254**] An output device **106** generates final output based on the input output data with or without further processing.

[**0255**] After receiving output data from server application **112**, the information apparatus **100** may then transmit the output data to the output device **106** selected by the user in step **806**. In some cases, the client application **102** may further process the output data (as shown in step **804**). This further processing in the information apparatus **100** may include one or more operations such as encoding, decoding, compression, decompression, rasterization, scaling, color correction, halftoning, watermarking, adding templates, name, time stamps, etc., as shown in step **804**, before sending the output data to the output device **106** through communication link **116** in step **806**. The client application **102** may involve other applications **103** for part or all of the processing in step **804**.

[**0256**] During the time when server application **112** processes a composite message, the communication link **116** between information apparatus **100** and output device **106** selected by the user may be constantly maintained or it may be temporarily dropped and then resumed for transmitting output data. Encryption techniques may be applied to the output data to ensure security. Compression/decompression, encode/decode may also be implemented to facilitate the transmission of output data over communication link **116**.

[**0257**] After receiving output data from information apparatus **100**, an output device **106** may simply buffer the output data, as shown in step **810**, before sending the data to output engine (or printer engine in the case of a printer) **1008** for final output. In cases where an output device **106** includes a printer controller **1010**, output controller **104** or combined controller, such controller may, jointly or individually, further process the output data (shown in step **814**) before sending the data to output engine or printer engine **1008** for final output.

[**0258**] It is possible that a plurality of information apparatuses **100** may transmit output data to an output device **106** at the same time or at least during overlapping periods. It may then become necessary to implement queuing or spooling functionality in the output device **106**. Preferably, the output device **106** stores output data it receives in a queue until the output data can be processed. Such queuing and spooling capability may be implemented partially or entirely by the output controller **104** as an optional feature set. A user may be notified when the output device **106** is ready to process a particular output job or the status of the output job in the queue. Once the output device **106** completes the output service, the user may terminate the output process by indicating such an intention through an interface on information apparatus **100** or by moving the information apparatus **100** out of the communication range of the output device **106**.

[**0259**] FIGS. **9A-9F** show a series of graphical user interfaces (GUIs) rendered at different times on a display screen **901** of an information apparatus **100** to illustrate one implementation of the output process **401** described with reference to **FIG. 4**. The illustrated example of user interface can be implemented in an information apparatus **100** with or without an operating system. The user interface is illustrated here only as an example, other user interfaces may also be implemented and in no way restrict the scope and claims of the present invention.

[**0260**] Referring to **FIG. 9A**, screen **901** may be a part of an information apparatus **100** such as, without limitation, palmtop, PDA, mobile phone, pager, Internet appliance, web pads, e-book etc. In this particular example, as shown in screen **901**, a user is reading a news article published on the Internet using a document browsing application in information apparatus **100**. Part of the article (e.g., photo **900**) cannot be fully displayed and so is indicated by a link or tag. The inability to fully display photo **900** may arise from a variety of reasons including limitations in display capacity, memory, processing power, or lack of support by a service provider, application, communication protocols, or software protocols, among other reasons.

[**0261**] As shown in **FIG. 9A, a** document browsing application in the information apparatus **100** may have a plurality of functions and feature sets available to the user through the GUI. A printing function control **902** may be one of these features. The output or printing functionality may be provided by accessing a client application **102** with or without an operating system. In this illustrated example, the user may invoke the output function by selecting the print function control or icon **902**. The user can make the selection by using, for example, any of a keyboard, keypad, mouse, stylus, soft keys, push buttons, software command, touch sensitive screen, voice-activated command, among others.

Other display methods or functionalities or feature sets such as a drop down list, a pull down menu, among others, are also possible.

[0262] When the user selects Print function control **902**, the client application **102** may be invoked or launched. The output process **401** described with reference to **FIG. 4** may then proceed. The client application **102** may obtain a document object (e.g., the news article and or the pointer of the new article) from the document browsing application. As described earlier, the user may be able to choose to output the reduced digital document (e.g., without photo **900**) as displayed in the information apparatus **100** or to output the full digital document in its original form (e.g., with photo **900**). We assume for this example that the user has preset the default to always output the full digital document in its original form.

[0263] Since the user does not have a pre-selected or default output device **106**, a discovery process **520** may automatically proceed in this example. A communication manager may coordinate the discovery process **520**. As described earlier, the communication manager may be part of the client application **102** or may be a separate application that can communicate with the client application **102**.

[0264] Assuming in this example that he or she did not specify any searching criteria for the discovery process **520**, the user may be presented with all the printers **106**, if any, that are available to take the print job. In the case where no available printers are found (not shown here), the user may also be notified and provided with alternatives as described earlier in the discovery process **520**.

[0265] Assume in this example that three printers **106** are found. Their identities and service charges are listed on the GUI illustrated in **FIG. 9B**. The user may select a "more information" control or icon **904** to get more detailed information on each available output device **106**. The user may select a printer **106** from the list, for example, by moving a cursor to highlight the name of the selected printer **106** and then selecting a "select" control or icon **906**. The user may also withdraw the service request from any of the three printers by selecting a "cancel" control or icon **907**.

[0266] Assume in this example that the user has selected the PH inkjet **260** printer. The next GUI, as illustrated in **FIG. 9**C, provides the user with a confirmation. Information included in the confirmation may include (1) address (pointers) of the digital document intended for output, (2) printer selected, (3) service fee that will be charged. Various payment types and methods may be used. The user may also choose to provide payment, deposit or escrow in various forms. However, for simplicity of illustration, only two are shown in this particular example. If payment by credit card is selected, the user may be prompted to provide credit card information. In one example, the user may then provide such information through a GUI (not shown) on the information apparatus **100**, which then transmits the information to the selected output device **106**. In another example, the user may provide credit card information directly to the selected output device **106** through an interface provide by the output device **106**. If the user has selected the cash payment option, cash may be physically deposited into a cash collector installed or connected to the selected output device **106**. Or the user may transfer cash in electronic form from the information apparatus **100** to the output device **106**. The user may or may not need to manually input payment information such as a credit card number. A user profile and or a variety of payment information may have already been saved in the user's information apparatus **100**. The client application **102** may automatically fetch necessary payment information and process the transaction with the output device **106** or with a remote transaction server. If the output service is free, no payment information will need to be collected. If the output service is a subscription, then membership, login, authorization, or security information may be collected instead or in addition.

[0267] After the user confirms all the information and provides payment or deposit as requested, the user may be prompted to describe the job object as illustrated in **FIG. 9D**. Some of the selections for the job object input illustrated here may be device-dependent while others may be device-independent. The user can make selections by using, for example, radio buttons among other controls. Some default values, such as items **908** and **909**, may have been preset by the user or automatically calculated by the client application **102** after negotiation with the output device **106**. The user may also change all or some default values. After describing the job object, the user may select the "Next" control or icon **910** to move to the next step. It should be noted that only some exemplary or commonly used selections are displayed in the exemplary GUI of **FIG. 9D**. The user may make further selections by, for example, selecting the "more option" control or icon **911** or by simply scrolling down the display.

[0268] After the user submits the job object, the client application **102** sends a composite message **430** to an application server **110**. The composite message **430** may include a document object, a printer object, and a job object. The application server **110** may contain server application **112**. In this example, the server application **112** may need to fetch the output content (in its original format) based on the pointers provided in the document object. The server application **112** may obtain or invoke helper applications to process the output content (e.g., the news article) into print data that can be accepted by the printer **106** selected by the user (e.g., PH Inkjet **260**). The server application **112** and or the helper application may perform one or more raster image processing operations on the output content. During this server application process **701**, the sever application **112** may optionally launch a GUI in a window **912** as shown in **FIG. 9E** to provide the user with estimated processing time and or status. The user may choose to hide this window by selecting the "hide" control or icon **914**. The user may also choose to abort the process by selecting the "cancel" control or icon **913**.

[0269] After it has finished processing the output content and has generated the necessary print data, the server application **112** transmits the print data to the selected output device **106** through the user's information apparatus **100**. Transmission may be wired or wireless. The client application **102** in the information apparatus **100** may or may not further process the print data before passing it to the output device **106**. The output device **106**, after receiving the print data, may queue the print job in its memory or storage component until it is ready to process this particular print job. As shown in **FIG. 9**F, the user may be notified when the original document is being printed or has been printed out. The user may also be notified of the service fee that has

been, for example, charged to a credit card or deducted from a cash deposit. The user may be provided with a "more option" control or icon **916**. For example, the user may choose this printer as a preferred default printer so that for the next output or print operation a discovery process **520** may be skipped or shortened if this printer **106** is found to be available.

[**0270**] FIGS. **9A-9F** provides only one example of output process **401** described with reference to **FIG. 4**. While the above descriptions contain many specificities, these should not be construed as limitations on the scope of the invention, but rather as exemplary embodiment thereof. Other user interfaces and embodiments may also be implemented. The process may also be implemented with more or fewer steps. As an example, security verification and authentication may be added to the steps illustrated in FIGS. **9A-9F**. As another example, the step illustrated by **FIG. 9B** may be skipped as the user may have pre-selected or preferred a default printer. These and other possible variations do not however depart from the scope and spirit of present invention.

[**0271**] Having described and illustrated the principles of our invention with reference to an illustrated embodiment, it will be recognized that the illustrated embodiment can be modified in arrangement and detail without departing from such principles. In view of the many possible embodiments to which the principles of our invention may be applied, it should be recognized that the detailed embodiments are illustrative only and should not be taken as limiting the scope of our invention. Rather, I claim as my invention all such embodiments as may come within the scope and spirit of the following claims and equivalents thereto.

We claim:

1. A data output device for rendering output content managed from an information apparatus, comprising:

a rendering engine that imparts output on a medium in accordance with output data acceptable to the output device; and

memory storage that stores at least part of one or more output device objects, with one or more attributes corresponding to the output device;

the output device having means for providing at least part of the output device object to the information apparatus, and means for receiving output data from the information apparatus, the output data related to the output content managed from the information apparatus and acceptable to the output device.

2. The output device of claim 1 further comprising means for storing one or more output device objects with one or more attributes corresponding to the one or more output devices.

3. The output device of claim 1 further comprising means for receiving plural service requests from plural information apparatuses, the output device providing the at least part of the output device object to each of the plural information apparatuses.

4. The output device of claim 1 in which the output device includes an audio output device.

5. The output device of claim 1 in which the at least part of the output device object is provided to the information apparatus in response to a service request received from the information apparatus.

6. The output device of claim 1 further includes means for providing the at least part of the output device object to the information apparatus as unsolicited information without a request therefor from the information apparatus.

7. The output device of claim 1 in which the at least part of the output device object is provided to the information apparatus in more than one communication session with the information apparatus.

8. The output device of claim 1 further includes means for implementing payment processing as compensation for rendering of the output content on the output device.

9. The output device of claim 1 further includes means for implementing job management functionalities with one or more of data output job queuing and spooling.

10. The output device of claim 1 further includes means for providing a user interface for receiving user-implemented modifications of the output device.

11. The output device of claim 1 further includes means for implementing a security procedure that limits access to the rendering provided by the selected output device.

12. The output device of claim 11 in which the security procedure includes storing in a memory component an access control list specifying an information apparatus for which the selected output device will render output content.

13. The output device of claim 11 in which the security procedure includes one or more of a subscription indicator, a login, a password, and an authentication.

14. The output device of claim 1 further comprising an output controller with means for communicating with the information apparatus.

15. The output device of claim 14 in which the output controller is separate from the output device.

16. The output device of claim 14 in which the output controller further includes means for converting the output data into a form acceptable to a printer controller.

17. The output device of claim 14 in which the output controller further including means for converting the output data into a form compatible with an output engine.

18. The output device of claim 1 in which the output data received by the output device includes device dependent data acceptable to the output engine.

19. The output device of claim 1 further includes means for converting the output data content into a form compatible with the output engine.

20. The output device of claim 1 in which the output device includes one of a display device or a projection device.

21. The output device of claim 1 in which the output device further includes means for performing at least partial raster image processing operations on the output data.

22. The output device of claim 21 in which the image processing operation includes one or more of an interpretation operation, a conversion operation, a rasterization operation, a scaling operation, a segmentation operation, color space transform operation, an image enhancement operation, a color correction operation, a halftoning operation, a compression operation, and an encryption operation.

23. The output device of claim 1 in which the output device includes a printer.

24. The output device of claim 1 in which the output data includes device dependent data in relation to the output device.

25. The output device of claim 1 in which the output medium is one or more of a substrate, a paper, a display screen, and a projection.

26. In a data output device for rendering output content managed with an information apparatus, the data output device including a rendering engine that imparts output on a medium in accordance with output data received at the output device, and a memory storage that stores data for rendering; the improvement comprising:

an output controller having means for providing to the information apparatus at least part of an output device object having one or more attributes corresponding to the output device, means for receiving output data related to the output content managed from the information apparatus, and means for passing the output data for rendering of the output content at the output device.

27. The output device of claim 26 in which the output data received at the output device is from a network.

28. The output device of claim 26 in which the output data received at the output device is from the information apparatus.

29. The output device of claim 26 in which the output data includes device dependent data in relation to the output device.

30. The output device of claim 26 in which the output medium is one or more of a substrate, a paper, a display screen, and a projection.

31. The output device of claim 26 in which the output controller further comprises means for storing one or more output device objects with one or more attributes corresponding to the one or more output devices.

32. The output device of claim 26 further comprising means for receiving plural service requests from plural information apparatuses, the output controller providing the at least part of the output device object to each of the plural information apparatuses.

33. The output device of claim 26 in which the output device includes a printer.

34. The output device of claim 26 in which the output controller further includes means for implementing payment processing as compensation for rendering of the output content on the output device.

35. The output device of claim 26 in which the output controller further includes means for implementing job management functionalities with one or more of data output job queuing and spooling.

36. The output device of claim 26 in which the output controller further includes means for providing a user interface for receiving user-implemented modifications of the controller.

37. The output device of claim 26 in which the output controller further includes means for implementing a security procedure that limits access to the rendering provided by the selected output device.

38. The output device of claim 26 in which the output controller is included in the output device.

39. The output device of claim 26 in which the output controller is separate from the output device.

40. The output device of claim 26 in which the output controller further includes means for converting the output data content into a form compatible with the output engine.

41. The output device of claim 26 in which the output controller further includes means for performing at least partially, raster image processing operations on the output data.

42. In a data output device for rendering output content managed with an information apparatus, the data output device including a rendering engine that imparts output on a medium in accordance with output data received at the output device, and a memory storage that stores data for rendering; the improvement comprising:

means for providing to the information apparatus at least part of an output device object having one or more attributes corresponding to the output device, means for receiving output data related to the output content managed from the information apparatus, and means for sending the output data to the output engine for rendering of the output content.

* * * * *