

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
14 August 2003 (14.08.2003)

PCT

(10) International Publication Number  
**WO 03/067400 A2**

(51) International Patent Classification<sup>7</sup>: **G06F**

(21) International Application Number: PCT/US03/03842

(22) International Filing Date: 7 February 2003 (07.02.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/355,227 7 February 2002 (07.02.2002) US

(71) Applicant and

(72) Inventor: **NICKERSON, Christian** [US/US]; 20  
Melville Road, West Windsor, NJ 08250 (US).

(74) Agents: **THIESSEN, Kendall, I.** et al.; Gibson, Dunn &  
Crutcher LLP, 1801 California Street, Suite 4100, Denver,  
CO 80202 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU,  
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU,

CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH,  
GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC,  
LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW,  
MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG,  
SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN,  
YU, ZA, ZM, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM,  
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),  
Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),  
European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,  
ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, SE, SI,  
SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN,  
GQ, GW, ML, MR, NE, SN, TD, TG).

**Declaration under Rule 4.17:**

— of inventorship (Rule 4.17(iv)) for US only

**Published:**

— without international search report and to be republished  
upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.



**WO 03/067400 A2**

(54) Title: ELECTRONIC WAITING ROOM

(57) Abstract: A system and method for monitoring and managing, in real-time, interactions between a healthcare provider, a patient and an insurer is disclosed. The system and method comprises of a virtual or electronic waiting room. Management and monitoring of patient interactions and transactions is achieved by storing attributes associated with a patient and linking these attributes to a virtual representation of the patient, which may be accessed and manipulated via the electronic or virtual waiting room.

## ELECTRONIC WAITING ROOM

System and method for electronically monitoring and managing, in real-time, interactions between a healthcare provider, a patient and an insurer. This application claims priority from U.S. Provisional Application No. 60/355,227 filed on February 7, 2002.

### BACKGROUND OF THE INVENTION

#### 1. Field of the Invention

The present invention relates to healthcare provider-patient-insurer interactions and more specifically to electronically managing and monitoring, in real-time, the information necessary for and the processing of such interactions.

#### 2. Description of the Related Art

Currently, health care providers (hereinafter referred to as the "Provider" or "Providers"), Patients (hereinafter referred to as the "Patient" or "Patients") and Insurers (hereinafter referred to as the "Insurer") need to exchange various pieces of paperwork in connection with any procedures performed or to be performed on the Patient and the insurance coverage for these proposed or completed procedures. Current systems for managing such transactions, which may be transacted either wholly or partially on paper, encounter the following problems: (a) recording and submission of accurate paperwork is necessary or transactions cannot be processed and must be resubmitted; (b) a particular request may require the submission of additional information that is in the custody of one of the parties – Patient, Provider or Insurer - but is not instantaneously available to all of the parties; (c) at the time of a Patient visit, the Provider may not know the status of a request for a particular procedure and, as a result, may risk performing a procedure that is not covered by insurance; (d) record keeping is unduly complex with each entity – Provider, Patient and Insurer – often keeping separate and incompatible records; (e) Providers and Patients have to fill out repetitive paperwork; (f) real-time management or oversight of Patient-Provider transactions is difficult; and (g) generally, real-time transactions between the Provider and Insurer are not possible or at best occur via telephone or facsimile.

Both Providers and Patients would benefit greatly from a real-time claims submission method that also provides an easy to use system for submitting insurance requests, receiving virtually immediate responses from Insurers and storing necessary information concerning Provider-Patient and Provider-Insurer transactions in a consistent form. What is needed, then, is a system and method for organizing the various Patients that are covered by a particular Provider, pooling necessary information from both the Provider and Insurer concerning these Patients, completing required information in a consistent manner, submitting coverage claims in connection with such Patients via the Internet, making Patient information available to multiple Provider employees, providing an interface for secure communications between a Provider and Insurer concerning a Patient and allowing Patients access to information concerning Provider-Insurer transactions.

#### BRIEF SUMMARY OF THE INVENTION

The following invention addresses the above-mentioned needs in the art by providing a Provider with a work management tool that allows the Provider to effectively manage and monitor a Patient's visit and any associated transactions in an organized and paperless manner. This tool can track a Patient from the time a Patient enters the Provider's office and insurance eligibility is verified through to the finalization of the Patient's coverage claim and applicable treatment with a few easy clicks. Similarly, a "Patient portal" is provided that provides the Patient with a summary of all activity relating to their requests for coverage and the status of those requests in a secure manner.

The Electronic Waiting Room includes the following key functions: (a) add Patient to Electronic Waiting Room; (b) remove Patient from waiting Room; (c) add work item(s) to Patient; (d) delete work item(s) from Patient; (e) save work item information; (f) sort Electronic Waiting Room; (g) change location of Electronic Waiting Room; (h) communicate work item request to Insurer; (i) receive communications relating to work items; (j) process and receive reports summarizing requests to Insurer and any responses from the Insurer; and (k) a Patient portal whereby a Patient can view insurance and procedure status. Utilizing these functions, any Patient of the Provider's staff who has been given access can track a Patient's status or add work items – even if the staff member is not in the same office as the one visited by the Patient. Also, by using standard work item formats, various information concerning a Patient can be filled in electronically by the Electronic Waiting Room system and, more importantly, the Electronic Waiting Room system can evaluate work items and

prompt corrections and additions as necessary. Furthermore, by incorporating the inventions described in the Legacy Data Conversion process as described in Appendix 1 and the System and Method for Managing Internet Transactions as described in Appendix 2, the Electronic Waiting Room can provide a Provider with real-time status of insurance coverage.

#### BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 is a flow diagram illustrating the creation of a patient profile for use in the Electronic Waiting Room.

FIG. 2 is a screen shot illustrating one method of organizing an Electronic Waiting Room interface.

FIG. 3 is a flow diagram illustrating how the Electronic Waiting Room may be used to process a Patient transaction.

FIG. 4 is a flow diagram illustrating how the Electronic Waiting Room may be used to manage Internet transactions with an Insurer.

FIG. 5 is a flow diagram illustrating how any employee of the Provider may access the Electronic Waiting Room and view patient status or modify or commence a transaction.

FIG. 6 is a flow diagram illustrating how the Electronic Waiting Room can be used to store information while a Patient transaction proceeds.

FIG. 7 illustrates how a Patient may access aspects of their profile remotely.

#### DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the construction and operation of preferred implementations of the present invention illustrated in the accompanying drawings. In those drawings like elements and operations are designated with the same reference numbers when possible.

The following description of the preferred implementations of the present invention is only exemplary of the invention. The present invention is not limited to these implementations, but may be realized by other implementations.

**FIG. 1** shows how, in a preferred embodiment of the invention, information from the Provider's Database **130** is pooled with information from the Insurer's Database **140** to create a Patient Profile **170**. This is a superior profile to those currently available to Providers because it contains information that the Insurer has selected as necessary for Insurer-Provider transactions concerning the patient.

The Electronic Waiting Room system and method commences when the Provider adds a new Patient to its system. One of the functions comprising the Electronic Waiting Room is a virtual waiting room, which is designated as such, and may reside as an electronic space, page or dedicated area on a Provider's system. **FIG. 2** illustrates how an Electronic Waiting Room interface **200** may be organized on the Provider's system. The system and method may use an Electronic Waiting Room interface **200** or any other embodiment to represent a waiting room, including the creation of a virtual waiting room. In a preferred embodiment of the invention, when a Provider accesses their computer system or network a visual icon establishes the existence of the Electronic Waiting Room interface **200** and when this icon is clicked on the Provider sees the virtual waiting room or Electronic Waiting Room interface **200**. The Electronic Waiting Room interface may be a web page maintained by the Provider or other electronic space on the Provider's system or network.

A new member may be added to the Provider's Electronic Waiting Room system by creating an Identifier **110** and entering various, necessary attributes **120**. The Identifier **110** and attributes **120** are stored in the Provider's Database **130**. Using the inventions described in the patent applications attached as **Appendix 1 and Appendix 2**, incorporated by

reference herein, the Provider can access the Insurer's Database **140**, via the Internet, without leaving the Electronic Waiting Room function, and confirm that the Patient is covered by the appropriate Insurer **150**. If the system does not show that the Patient is covered by the appropriate Insurer **150** manual intervention is necessary and the Provider may contact the Insurer for coverage approval. If, however, insurance coverage is confirmed, information selected by the Insurer and stored on the Insurer's Database **140** is added, via the Internet, to the Patient's attributes **120** stored on the Provider's Database **130**. In this way, necessary information concerning the Patient in the custody of the Provider **120** and Insurer **140** is pooled to create a complete and accurate Patient Profile **150** for Provider-Insurer purposes. This method is superior to the state of the art because it provides the Provider, in real-time, with additional information concerning the Patient that can be used to determine appropriate treatment and available insurance coverage. Furthermore, this complete Patient profile increases the probability of successful future Provider-Insurer transactions.

**FIG. 3** illustrates how a Provider-Patient transaction occurs using the Electronic Waiting Room. Once a Patient Profile **150** has been created and stored in the Provider's Databases **130**, the Provider can select the Patient's information by entering the Patient's Identification **110** - i.e., the Patient's social security number or insurance number - or via a Patient search. Once the Provider selects a Patient, an electronic representation of the Patient appears - i.e., a name or insurance number - on the Provider's portal and this representation can be moved by the Provider into the Electronic Waiting Room **320**. In the preferred embodiment of the invention, the Patient's representation is an icon containing the Patient's Identification that can be dragged over and into the Electronic Waiting Room feature on the Provider's portal. The Patient icon is linked to a page that includes various attributes of the Patient that may include personally identifiable information as well as information relating to their prior medical history **170**.

Once a Patient is added to the Electronic Waiting Room **320**, the representation of the Patient and the linked Patient attributes are accessible to any employee of the Provider who has been given access to the Electronic Waiting Room. In the preferred embodiment of the system, a receptionist enters a Patient's identification **310** and moves the electronic representation of the Patient to the Electronic Waiting Room **320**. At this point, the Patient representation and linked information is available to any of the Provider's employees who have an access code. In a preferred embodiment of the invention, employees can be given different access levels – i.e., a nurse's access may be limited to retrieving a Patient's medical history or an office manager's access may be limited to a Patient's insurance information. In a preferred embodiment of the invention, the Provider's offices contain numerous computers with access to the Provider's system and the Electronic Waiting Room or staff members are provided with handheld units linked to the Provider's computer system.

By accessing the Electronic Waiting Room **320**, the Provider can select a Patient from the Waiting Room and pull up information concerning the Patient or select a 'work item' **330** ('work item' is used to describe any transaction – i.e., creating a claim or a pre-certification), for example, a nurse may access the Patient profile to look at the Patient's medical history, a doctor may also access the work history and may also add a diagnosis and proposed treatment as a Work Item, or an administrative employee may add a request for insurance as a Work Item. Similarly, an administrator can navigate to a centralized Patient home page that allows the Provider to edit the stored Patient profile. In this way, any new information provided by a Patient can be added to the Patient profile.

In a preferred embodiment of the invention, Work Items can be pulled down from a menu **330**, for example, a doctor may complete a "diagnosis" Work Item or an administrator may complete a 'request for coverage for treatment' Work Item. In this way, the Provider can

use standard forms and necessary information in the Patient's profile can be automatically inputted into the forms removing the redundancy of repeatedly entering the same Patient information into different forms. Also, standard forms can have prompts so that, for example, when a Patient's required treatment is described on the form the treatment prompt ensures that the treatment description meets an Insurer's approved treatment descriptions. In this way, errors, lack of required information etc are less likely to occur and Provider-Insurer transactions are less likely to be rejected as requiring more or Insurer compliant information. Furthermore, by using the Patient Profile 170, information obtained from the Insurer can be added to the standard forms and, consequently, the resulting Work Items 340 are likely to be more accurate and to meet Insurer requirements when submitted 350.

Once the Provider selects the Patient 's representation and moves it to the Electronic Waiting Room 320, he or she can then add a Work Item for that particular Patient 330. The Provider has the option of entering information about the Work Item 340 or performing other work within the portal and then returning to the original Work Item and completing the Work Item; the information for a Work Item is saved for a Patient until it is either deleted or completed. Once the Provider completes the information on the Work Item 340, they can submit the item 350 and it disappears from the Electronic Waiting Room. Optionally, as discussed previously, different Providers (or staff Patients of the Providers) may have rights to perform different transactions with respect to the Patient, for example, a bookkeeper could submit a claim for payment on a procedure while only a nurse could actually create an entry for the new procedure. The various realm of actions that a particularly employee of the Provider may perform may be defined by the Provider, by the Patient, or by the Insurer.

Once a Work Item has been submitted 350, the Provider receives a real-time update of the Work Item status 360. The Work Item update 360 may show the change in the Patient



Profile submitted by the Provider or it may show the status of a request for insurance coverage (See FIG. 4). Once the Work Item has been submitted, the Patient Profile 170 is updated to show the relevant changes 370. In this way, all Provider-Patient transactions are recorded in the Patient Profile 170 in real-time.

Finally, once the Provider-Patient transaction is completed, the Patient is removed from the Electronic Waiting Room 380.

FIG. 4 shows a preferred embodiment of the Electronic Waiting Room invention. In this embodiment the inventions disclosed in the attached patent applications, **Appendix 1 and Appendix 2**, are incorporated by reference. By incorporating these inventions, it is possible for a Provider to complete a Work Item requesting insurance coverage 410 from an Insurer and receive in real time, via the Internet, a response from the Insurer 420. If the real-time response establishes that coverage was granted then the Patient Profile 170 is updated and the Provider can perform treatment with the knowledge that the Insurer will cover the treatment. If, however, coverage is denied 330, the Provider can edit the request 310 and resubmit the request 310. If after editing, the request is once again denied then manual intervention by the Provider is necessary.

In this way, not only does the Electronic Waiting Room provide a management tool for Patient transactions it also provides a method for seamlessly and in real-time requesting and receiving insurance coverage adjudications.

FIG. 5 illustrates how the Electronic Waiting Room provides accessible, real-time information to any member of a Provider's staff whatever the staff member's location. The staff member (hereinafter referred to as the "User") begins a session 510 by navigating to the Provider's portal. The User then enters his or her identification and if the identification is valid, the User is given access to the Electronic Waiting Room 530. The User can then view

the Patient Profile or any Work Items that have not been completed **540**. In this way, the Patient's status is available to any member of the Provider's staff with access to the Electronic Waiting Room and Work Items can be viewed and edited by multiple staff members **550**. Once a User has viewed/edited the Work Item, it is then submitted in the normal manner **350** and the User exits the Electronic Waiting Room **560**.

This functionality described in **FIG. 5**, allows the Provider to assign/reassign Work Items to any staff member in the office with access to the Electronic Waiting Room. In a preferred embodiment, this can be done by selecting the Electronic Waiting Room manager icon and assigning the Work Item to someone else in the office. The new owner will receive a message explaining that the Work Item was reassigned to them. If the Provider desires to take ownership of a particular Work Item, they can select the Work Item and then select 'Take ownership'. A message will be sent to the original owner explaining someone else now owns the Work Item.

Furthermore, when Providers have multiple offices, the Provider and Patient portals can serve as a "virtual" office manager. For example, if a Provider wishes to know whether a Patient has come for an appointment, the Provider can switch 'locations' by selecting a different location above the Waiting Room. This allows a Provider to monitor Work Items in another Provider office and provide remote support in helping to close out various Work Items.

**FIG. 6** illustrates how the Electronic Waiting Room allows the retention of Work Items. The User begins a session **510**, selects a Patient **310**, moves the Patient to the Electronic Waiting Room **320**, selects a Work Item **330** and begins inputting the necessary information **340**. At this point, the User is free to handle other matters or gather more information since the Work Item and inputted information is stored with the Patient's Profile

170. In this way, a User can simply return and finish a Work Item later or leave the incomplete Work Item to be completed by another User 610. As a result, the Electronic Waiting Room provides a simple management tool that reduces paperwork and stores incomplete transactions. In a preferred embodiment of the invention, a reminder function alerts Users to incomplete Work Items.

FIG. 7 illustrates how the invention may be used by a Patient to view the status of transactions. The Patient 710 begins a Web Session 720 and navigates to the Provider's portal. The Patient enters their Identification 730 and if it is valid the Patient can view the status of Work Items and attributes of their profile 740 that the Provider has selected for Patient viewing.

Similarly, the Electronic Waiting Room can include "inline" messaging system as a means for communicating with Patients. For example, if a particular Patient is scheduled for a particular procedure, that Patient can be notified when that procedure has been pre-certified. By using an "inline" messaging system through a browser, this information can be securely communicated to the Patient without the need for additional software or training, such as is often required with secure electronic mail correspondence.

The Electronic waiting Room has many advantages over current state of the art Provider-Patient transaction methods, including:

- Flexibility to save information within a 'form' before submission
- Ability to assign specific 'tasks' to other people in the Provider office virtually.
- Online, interoffice notification capability to assign and reassign Work Items to other people in the office.

- Workflow integration throughout the office to monitor workload of each of the individuals within a specific office location.
- Remote Provider office support to complete Work Items in other office locations.
- Ability to take ownership of uncompleted Work Items to expedite payment processing in a specific office location.
- Secure communications to Patients through use of a portal interface.
- Increased efficiency resulting from reduced Patient contacts concerning claim status etc.

The foregoing description of the preferred embodiments of the invention has been presented for the purposes of illustration and description. The descriptions of the header structures should not be limited to the embodiments described. For these reasons, this description is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

## METHOD AND SYSTEM FOR CONVERTING LEGACY DATA

Inventor: Craig Mulligan

Method and System for converting legacy data into one or more useable forms including, without limitation, extensible markup language (XML). This application claims priority from provisional application serial number 60/355,224 filed on 2/7/2002.

### BACKGROUND OF THE INVENTION

#### 1. Field of the Invention

The invention relates to a process for converting data and specifically to the process of converting legacy data into a format that can be viewed via the Internet.

#### 2. Description of the Related Art

Currently, it is not uncommon for insurance, aerospace, retail, banking manufacturing, communications, health care, and financial companies to provide an interface, via the Internet, for providing access to a user's data. The problems associated with enabling this access include: (a) much of the data on a user is stored in legacy systems; (b) legacy systems often require very specific and proprietary processing instructions; and (c) once the data is retrieved, the data often needs to be placed in a useable format.

Many information systems being used today in certain established industries, insurance, aerospace, retail, banking manufacturing, communications, health care, and financial companies, use a system called the Information Management System or IMS (some estimates place ISM penetration at close to 90% in these critical industries). IMS is comprised of two primary components: a transaction management system and a hierarchical database management system. The IMS transaction manager includes a facility called an open transaction manager access facility or OTMA. OTMA provides an easier way of

### APPENDIX 1

enabling users that connect to a system using the TCP/IP protocol to make information requests to IMS. OTMA further includes a cross-system coupling facility (XCF) that facilitates communications between OTMA and OTMA clients.

The process of delivering data to web applications in the prior art was to first invoke a transaction (submit a transaction request) through a message queue (MQ) series, The requested transaction is then passed to OTMA, which in turn passes it to the IMS for processing by applications being managed by IMS. Once the transaction is complete, a reply is passed back through OTMA and eventually back to MQ. MQ then strips off any data that was included in the request (the MFS overhead) and the data is delivered to the middleware layer. Thereafter, either the middleware layer (or the underlying web application) receives the data and places it into a useable format.

As can be appreciated, this process involves several steps resulting in both a longer lag time prior to delivery to the web application and requires multiple transaction requests that creates significant overhead (such as MFS overhead) on the overall system. What is needed, then, is a system in which a transaction would be invoked using the same software tools in a single request and which could deliver the data into a useable format without requiring the same MFS overhead.

#### BRIEF SUMMARY OF THE INVENTION

The following invention addresses the above-mentioned needs in the art by providing a method for submitting and processing data requests without requiring any extra overhead wherein the data requested by the transaction is delivered in an XML format. The method begins when the IMS control region evokes one of the legacy transactions, passing the I/O area containing the message to the transaction. The control region is the execution environment for the IMS system software, the control blocks and storage pools required for

managing communication access and application program scheduling. In the preferred embodiment, the transaction passes the I/O area to a program, such as E900010. E900010 scans the I/O area and breaks out each of the key words passed in the I/O area. If no key words have been passed or identified by E900010, the transaction ends and a code indicating the no key words were present is returned. If key words were passed, control is temporarily passed to the database portion of IMS server to determine if the requested data is housed within the database. If there is no data for the request, a code indicating this fact is returned. If data is detected, the transaction continues by retrieving the driver parameters. In the preferred embodiment, the driver parameters are retrieved from the database using the E9000DBP module provided with IMS. These parameters are then placed into a data array for processing.

Once the data keys have been set and the driver parameters stored, control is passed to an XML object processor. Using pointers to the data array, the system begins processing the "object." While processing the parameter cards, which are the XML object structure, if an embedded tag is encountered, the program passes control to a subsequent copy of itself, which then processes the embedded object. The parameter cards for each object may be used to invoke further database calls to retrieve any additional data required to fulfill the object. As each process completes, the "prior" application is run until the object and all associated data have been retrieved and assembled.

This recursive process (or emulated recursive process) is normally not permitted by Cobol but the process above avoids this restriction. Additional functions may also be used. For example, E900UCNV may be driven by the parameter cards to clean any character strings in the data that resides on the database on internal processing area. E9000LMS may be used to prefix and postfix each piece of data with the tag housed within the parameter

cards. The data is subsequently processed into the I/O area, which is then returned to the requestor. Finally, E9DSPLY may be used for debugging purposes.



## BRIEF DESCRIPTION OF THE FIGURES

Figure 1 is a block diagram illustrating a standard IMS system including OTMA and IMS connections.

Figure 2 shows a schematic/block diagram of a computer system.

Figure 3 is a block diagram illustrating the flow of data in one embodiment of a computer system logically connected to the system and method of the present invention.

Figure 4 is a flowchart illustrating how a transaction request is submitted to the IMS server.

Figure 5 is a flowchart illustrating the improved transaction processing and data conversion process of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the construction and operation of preferred implementations of the present invention illustrated in the accompanying drawings. In those drawings like elements and operations are designated with the same reference numbers when possible. The following description of the preferred implementations of the present invention are only exemplary of the invention. The present invention is not limited to these implementations, but may be realized by other implementations.

Many information systems being used today in certain established industries, such as financial, banking and insurance providers, use a system called the Information Management System or IMS. Referring now to FIG. 1, a block diagram illustrating a standard IMS system 100 that may be used in accordance with the present invention is shown. IMS 100 is comprised of two primary components: a transaction management system and a hierarchical database management system (not shown). The IMS system 105 includes a facility called an open transaction manager access facility or OTMA 110. OTMA 110 provides an easier way of enabling computer system 120 using a transmission control protocol/internet protocol (TCP/IP) connection 125 to make information requests to IMS server 105. OTMA 110 further communicates with a cross-system coupling facility (XCF) 130 that facilitates communications between OTMA 110 and OTMA clients 140. Alternatively, an IMS terminal 150 may also be used to communicate and otherwise submit requests to the IMS Server 105.

FIG. 2 shows a basic computer system 120. The computer system 120 contains a Processing Element 202. The Processing Element 202 communicates to other elements of the Computer System 200 over a System Bus 204. A Keyboard 206 allows a user to input information into Computer System 200, and a Graphics Display 210 allows Computer

System 200 to output information to the user. Graphical Input Device 208, which may be a mouse, joy stick, or other type of pointing device, is also used to input information. A Storage Device 212 is used to store data and programs within Computer System 200. A Memory 216, also connected to System Bus 204, contains an Operating System 218, and relevant Software 220. A Communications Interface 214 is also connected to System Bus 204. Communications Interface 214 may have one or more serial ports, parallel ports, infrared ports, and the like. Connectable through Communications Interface 214 may be an external printer or scanner, as well as access to a computer network (LAN or WAN), to the Internet, or to any other appropriate communication channel.

Referring now to FIG. 3, a sample embodiment of a computer system 120 running one or more web based applications and interacting with the IMS server is shown. In this example, the web-based applications include a web browser. In the preferred embodiment, the web browser supports the extensible markup language or XML. Using this browser, a request is submitted for data and that request is submitted via TCP/IP 125 to a web server 310. The web server 310 may further include various CGI programs and/or scripts 320 that receive messages from the web browser and processes the transaction. As explained above, in the prior art, the process of delivering data to web applications that are running on a computer system 120 begins by invoking a transaction through a message queue (MQ) series (not shown). The requested transaction is then passed to OTMA client 140 and OTMA 110, which in turn passes it to the IMS 105 for processing. Once the transaction is complete, a reply is passed back through OTMA 110 and eventually back to MQ. MQ then strips off any data that was included in the request (the MFS overhead) and the data is delivered the CGI program 320. The CGI program 320 then delivers it back to the web server 305 that transmits the final result via TCP/IP 125 for display on the computer system 120.

For example, let us assume that the requesting agent enters information into the computer system 120 and submits a request to update their address via a web server 310. In the medical-insurance industry, the requesting agent may be a physician's office, medical center, hospital etc and the relevant software 220 in the computer system 120 might be the Physician Office and Management System. In a preferred embodiment of the invention, the Web server 310 reviews the information entered by the requesting agent in real time for embedded viruses and errors. Also, web server 310 may also verify the authority of the requesting agent to access the requested transaction. In a preferred embodiment of the invention, the formats and field sizes of all data elements of the information entered by the requesting agent are analyzed by the web server 310 and mis-keyed or erroneous data is brought to the attention of the requesting agent. In this way, any submitted request with incompatible data is halted and a notice highlighting the incompatible data is returned to the requesting agent. In the preferred embodiment of the invention, the web server 310, in conjunction with one or more CGI programs 320, creates a string of the data submitted by the requesting agent and this string is converted through a real-time transaction into the appropriate entry format for the OTMA client 140.

Thereafter, a transaction record is created by storing one or more attributes relating to the requesting agent's transaction into a single entry having a locally unique identifier. The identifier could be a database entry address, a pointer to memory location, an offset value, or other methods that can be used to establish a locally unique identifier for the stored data. Not all attributes need necessarily be stored, only attributes relating to the requested host or process. In the preferred embodiment of the invention, the attribute or attributes relating to the requested host process are stored in a database that is associated with or under the control of the IMS server 105 and given a unique identification. For example, the attributes could be assigned a unique ID by storing separate instances within the same database. In the preferred

embodiment of the invention, an identifier associated with the requesting agent or the person for whom the request is made; i.e. home address or telephone number, is stored in the Database.

In the middleware, an initial tier of the IMS server (or associated programs or routines) prepares and formats the requesting agent's submitted transaction for transmission to a host (not shown). While this invention will be described with reference to managing an assortment of transactions, the invention could also be applied in such a way that only related transactions are managed by the same transaction manager. By way of example, the initial tier of the system may use Enterprise Java Beans ("EJB") to access the database associated with a host. (Java Beans is a trademark of Sun Microsystems, In.). As persons familiar with the art are aware, a Java Bean can be used to access data or applications that create data, from a legacy host or database.

The Transaction Manager of the IMS server 105 transmits the formatted transactions to a host. In the preferred embodiment, a host is a unit that includes at least one processor and one or more instructions associated with a process performed by the host. For example, a host may be an "update host" that updates entries to a particular database. While the word host implies singularity, the "host" unit may comprise one or more actual processing units that may be physically separated or otherwise distributed on a given network. Finally, the designation of a processing unit as a "host" in one transaction does not restrict the host from serving as an initiating agent in the same or related transactions. For example, a given host may submit requests to one or more host "subprocessing" machines. As a result, a single requested transaction could result in transaction entries for different layers of processing. Alternatively, there could be only a single transaction record that includes each host that will be used (whether directly or as a "subprocess" host) to complete the requested transaction.

When the host is available, it receives and can commence processing the transaction request sent by the requesting agent via the middleware bridge (if applicable).

Referring now to FIG. 4, a flow chart illustrating the method of the present invention is shown. The method of the present invention begins when the IMS server 120 receives the transaction request and the control region of the IMS Server 120 evokes 405 one of the legacy transactions, passing the I/O area containing the message to the transaction. As explained above, the control region is the execution environment for the IMS system software, the control blocks and storage pools required for managing communication access and application program scheduling.

In the preferred embodiment, the transaction passes 410 the I/O area to a program, such as E900010. In the preferred embodiment, E900010 breaks out or otherwise identifies 415 each of the key words passed in the I/O area. Again, the specific manner in which this is performed is not relevant to the present invention provided that the program being used is capable of identifying one or more key words in the transaction request. If no key words have been passed or identified by E900010, the transaction ends and returns 420 a code indicating the no key words were present. If key words were passed, control is temporarily passed to the database portion of IMS to determine 425 if the requested data is housed within the database. If there is no data for the request, the system returns 430 a code indicating this fact. If data is detected, the transaction continues by retrieving 435 the driver parameters. In the preferred embodiment, the driver parameters are retrieved from the database using the E9000DBP module provided with IMS. These parameters are then stored 440 in a data array for processing. Once the data keys have been set and the driver parameters stored, control is passed 445 to an XML object processor.

Referring now to FIG. 5, a flow chart illustrating the XML processing of the present invention is shown. Using pointers to the data array, the system begins processing 505 the "object" including both the data keys and parameter cards. While processing 505 the parameter cards, which are the XML object structures, the program determines if an embedded tag is present 510. If not, the program continues to process 505 the object accordingly. If an embedded tag is encountered, the program creates or calls another instance of the program 515 and passes control 520 to the subsequent copy of itself. This "copy" or additional instance then processes the embedded object. The parameter cards for each object may be used to invoke further database calls to retrieve any additional data required to fulfill the object. As each process completes 530, the "prior" application is run until the object and all associated data have been retrieved and assembled. While the flowchart only illustrates the processing of a single embedded tag, it is quite possible for the embedded tag to itself include embedded tags. In such a case, another instance of the program would be created to process this "second level" embedded tag. Again, this process can continue to call additional copies of the program until the last of the embedded links have been resolved.

This recursive process (or emulated recursive process) is normally not permitted by Cobol (the base language used in IMS) but the process above avoids this restriction. Additional functions may also be used. For example, a program or call, such as a call to E900UCNV, may be driven by the parameter cards to clean any character strings in the data that resides on the database on internal processing area. A program or call, such as E9000LMS, may be used to prefix and postfix each piece of data with the tag housed within the parameter cards. Finally, a program or call, such as E9DSPLY, may be used for debugging purposes. Once the data has been fully processed it is returned to the I/O area, which is then returned to the requestor.

The present invention provides a number of important benefits. First, flexibility is achieved by allowing the addition of parameter cards. If a parameter card is added, additional data is sent to the web server 310 within the XML object with little to no additional coding. If the data has already been retrieved, all of the data is available to be sent. Second, this approach that the programs are reusable. This is because, in the preferred embodiment, most of the work modules may be called dynamically. This allows the same code to be used by multiple transactions. This also allows for a uniform flow through each of the transactions resulting in easier production maintenance.

Second, the speed of processing is increased because multiple steps are eliminated. The first step that is eliminated is the work done via MQ series calls. In the simplest case, the work of stripping off the additional data sent with traditional transactions tacked on by MFS is eliminated, for example. Additionally, since the transaction was built from scratch meaning that MFS was not used, the transaction is allowed to directly respond back to the requester with an XML object.

Finally, traditional transactions are built to serve inquiry processing meaning that the inquiry often requires that multiple transactions be invoked to satisfy a submitted request. By having a stand-alone web transaction, the system only invokes one transaction instead of 15 to 25 transactions. This would often have be the case when using the E9SVLD of the IMS server 105 for example. The step of reformatting the responses from the multiple transactions has been eliminated because the response is in an XML format, which can be used immediately by the web application.

The foregoing description of the preferred embodiments of the invention has been presented for the purposes of illustration and description. For example, although described with respect to a particular set of functions provided by IMS, any number of different



functions that provide similar capabilities could be used to implement this invention.

Additionally, the claimed system and method should not be limited to the particular system and met work configurations disclosed. For these reasons, this description is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

**SYSTEM AND METHOD FOR MANAGING INTERNET TRANSACTIONS**

Inventor: Shevin Conway

System and method for performing seamless transactions over the Internet with real-time feedback as to the status of transmitted transaction requests. This application claims priority from provisional application serial number 60/355,226 filed on February 7, 2002.

**BACKGROUND OF THE INVENTION****1. Field of the Invention**

The invention relates to Internet transactions and more specifically to managing and monitoring transaction requests sent via the Internet to a host.

**2. Description of the Related Art**

The access to information and communications, provided by the Internet, has fueled its explosive growth. Documents, messages and information that would previously have been sent via postal mail, over night delivery, or facsimile are now sent over the Internet. The ability to instantaneously communicate over the Internet has forever changed business practices.

Modern businesses have recognized that the Internet is not only a means of sending information; it can also be a tool for processing transactions. In the most basic situation Internet users contact each other by email and from the information shared can transact with each other. If, by chance, both parties are online at the same time the parties can transact with each other in real-time. On a slightly higher level, consumers can now go to virtual stores and purchase items via virtual shopping transactions. In this situation, the consumer views a company's available products over the Internet and by filling out the appropriate information the consumer can purchase desired items. The company selling the products receives the customer's request fills the order and emails the consumer a confirmation of the

order. Virtual shopping shows how at a most basic level the Internet can be used for transaction processing.

On a more complex level, it is now common for businesses to maintain Internet hosts, which are designed to perform real-time processing of transaction requests received via the Internet. In a common situation, a client of the company, an end-user, enters information into a data terminal. Most data terminals are personal computers and the client information could be entered onto the data terminal via a company web page or through a software management system. The data terminal is in turn connected to the Internet through a web portal. The information submitted by the client is transferred from the data terminal to the company's host ("Company-Host") via the Internet. The Company Host receives and then processes the end-user's data. After processing the data, the Company Host communicates a result to the end-user via the Internet. Such a transaction between an end-user and host occurs in real-time without the need for manual intervention. Consequently, the Internet provides a simple and efficient method for performing real-time transactions.

Several problems currently exist that reduce the effectiveness of Internet transactions. Many companies have applications and data stored on older legacy systems, which cannot be effectively accessed from the Internet. Another common problem with transactional methods that utilize the Internet is host unavailability. A company's host may be off-line to an end-user because of connection problems, limited capacity to process concurrent requests or various other reasons. In such situations, the host may send a message informing the end-user that the transaction cannot be processed or, in the worse case scenario, the transaction is not processed and the end-user is unaware of the failure of the transaction.

In many fields of business, including the insurance industry, it is common for an end-user to enter a large batch of individual transactions at one time. In such situations, if the

applicable hosts are unavailable and do not process some of the transactions, the end-user has the responsibility of determining which transactions were processed and resubmitting unresolved transaction requests. Similarly, under the current state of the art, an end-user that is continuously generating information for processing by a host has to independently keep track of the information that has been submitted and successfully processed to ensure processing of all transactions. Under the current state of the art, end-users submitting multiple transaction batches or continuous transaction requests only receive feedback if the transaction is either successful or unable to be processed by the host. This leaves long periods of time when the end-user is unsure of the status of a transaction. Additional problems associated with Internet transaction systems occur when an end-user provides too little, or incorrect information to the host and, as a consequence, the requested transaction cannot be processed. These problems are detrimental to the end-user who has to spend time manually resubmitting transaction requests that have failed because of host unavailability and assessing the status of transaction requests to ensure all requests in a batch etc are processed.

Consequently, there is a need, especially in the insurance industry, for a method of managing end-user-host transactions so that transactions requests are seamlessly delivered to the host for processing and real-time status reports concerning individual requests are provided to the end-user.

#### BRIEF SUMMARY OF THE INVENTION

The following invention addresses the above-mentioned needs in the art by providing a method for an end-user to seamlessly submit information from their data terminal via the internet, have the information processed/adjudicated by a host and receive real time status regarding the submitted information.

Specifically, the claimed invention stores an attribute from each transaction request submitted by an end-user. An attribute may comprise of the entire transaction request, an identifier associated with the transacting agent – i.e. a web address – or it could be a requirement of the transaction – i.e. needs system available or the transaction data needs. In turn, each stored attribute is given a unique identification. The storing and identification of attributes occurs in a middle tier. Once an attribute from a transaction has been stored and identified, a request is sent to the company host via the middleware. A polling process then begins periodically checking the status of all requests. If the polling process detects that a request has successfully been processed by the company host it is considered finalized and a message is sent to the end-user along with the result of the adjudication. If the request is not finalized because of host unavailability the request is stored and resubmitted when the host is back online. If, however, the host is unable to adjudicate the request because of problems other than unavailability an error report is generated and sent to the end-user to prompt external intervention.

In this way, the method provides the end-user with real-time feedback regarding the status of their request. The process also stores requests for resubmitting to the host, taking this task and the associated problems out of the hands of the end-user. In summation, the method provides a method for providing seamless Internet transactions for an end-user.

#### BRIEF DESCRIPTION OF THE FIGURES

Figure 1 is a block diagram illustrating a preferred embodiment of the present invention.

Figure 2 shows a schematic/block diagram of a computer system.

Figure 3 is a flow diagram illustrating the flow of information between the end-user and an online host in a preferred embodiment of the invention.

Figure 4 is a flow diagram illustrating the flow of information between an end-user and an offline host in a preferred embodiment of the invention

Figure 5 is a flow diagram illustrating side-by-side comparisons of the flow of a transaction through a preferred embodiment of the invention

Figure 6 illustrates the processes occurring as a transaction is processed in the different layers of a preferred embodiment of the invention

#### DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the construction and operation of preferred implementations of the present invention illustrated in the accompanying drawings. In those drawings like elements and operations are designated with the same reference numbers when possible.

The following description of the preferred implementations of the present invention is only exemplary of the invention. The present invention is not limited to these implementations, but may be realized by other implementations.

FIG. 1 is a functional block diagram showing a system **100** for synchronous and asynchronous transaction management with real-time feedback of transaction status in accordance with the present invention.

By way of example, a computer system **200** is connected to the internet via a web portal **120**. FIG. 2 shows a basic computer system **200**. The computer system **200** contains a Processing Element **202**. The Processing Element **202** communicates to other elements of the Computer System **200** over a System Bus **204**. A Keyboard **206** allows a user to input information into Computer System **200**, and a Graphics Display **210** allows Computer System **200** to output information to the user. Graphical Input Device **208**, which may be a mouse, joy stick, or other type of pointing device, is also used to input information. A

Storage Device 212 is used to store data and programs within Computer System 200. A Memory 216, also connected to System Bus 204, contains an Operating System 218, and relevant Software 220. A Communications Interface 214 is also connected to System Bus 204. Communications Interface 214 may have one or more serial ports, parallel ports, infrared ports, and the like. Connectable through Communications Interface 214 may be an external printer or scanner, as well as access to a computer network (LAN or WAN), to the Internet, or to any other appropriate communication channel (not shown in FIG. 1).

The requesting agent enters information into the computer system 200 and submits a transaction request via a web portal 110. In the medical-insurance industry, the requesting agent may be a physician's office, medical center, hospital etc and the relevant software 220 in the computer system 200 might be the Physician Office and Management System ("POMIS"). In a preferred embodiment of the invention, the Web Portal reviews the information entered by the requesting agent in real time for embedded viruses and errors. Also, Web Portal verifies the authority of the requesting agent to access the Host. In a preferred embodiment of the invention, the formats and field sizes of all data elements of the information entered by the requesting agent are analyzed by the Web Portal and mis-keyed or erroneous data is brought to the attention of the requesting agent. In this way, any submitted request with incompatible data is halted and a notice highlighting the incompatible data is returned to the requesting agent.

In the preferred embodiment of the invention, the Web Portal creates a string of the data submitted by the requesting agent and this string is converted through a real-time transaction into the appropriate entry format for the Host 150. A secure Internet protocol is utilized to deliver the data string to the Host 150. The system and method may utilize a fail-

over mechanism. For example, a string image of every transaction request could be stored in a database.

In the preferred embodiment, the computer system is connected to the Host 150 by a middleware bridge. A transaction record is created by storing one or more attributes relating to the requesting agent's transaction into a single entry having a locally unique identifier. The stored attribute may be an identifier associated with the transacting agent – i.e. a web address – or it could be a requirement of the transaction – i.e. needs system available or the data needs. The identifier could be a database entry address, a pointer to memory location, an offset value, or other methods that can be used to establish a locally unique identifier for the stored data. Not all attributes need necessarily be stored, only attributes relating to the requested host or process. In the preferred embodiment of the invention, the attribute or attributes relating to the requested host process are stored in the Database 130 and given a unique identification. The fact that the attributes are stored in separate instances of the same database guarantees this uniqueness. In the preferred embodiment of the invention, an identifier associated with the requesting agent or the person for whom the request is made; i.e. a social security number, name, address or telephone number, is also stored in the Database 130.

In the middleware, an initial tier of the system 120 prepares and formats the requesting agent's submitted transaction for transmission to the Host 150. While this invention will be described with reference to managing an assortment of transactions, the invention would also be applied in such a way that only related transactions are managed by the same transaction manager. By way of example, the initial tier of the system 120 may use Enterprise Java Beans ("EJB") to access the database associated with the Host 150. (Java Beans is a trademark of Sun Microsystems, Inc.). As persons familiar with the art are aware, a



Java Bean can be used to access data or applications that create data, from a legacy host or database.

The Transaction Manager **140** transmits the formatted transactions to the host **150**. A host is a unit that includes at least one processor and one or more instructions associated with a process performed by the host. For example, a host may be an "update host" that updates entries to a particular database. While the word host implies singularity, the "host" unit may comprise one or more actual processing units that may be physically separated or otherwise distributed on a given network. Finally, the designation of a processing unit as a "host" in one transaction does not restrict the host from serving as an initiating agent in the same or related transactions. For example, a given host may submit requests to one or more host "subprocessing" machines. As a result, a single requested transaction could result in transaction entries for different layers of processing. Alternatively, there could be only a single transaction record that includes each host that will be used (whether directly or as a "subprocess" host) to complete the requested transaction.

When the Host **150** is available, it receives and can commence processing the transaction request sent by the requesting agent via the middleware bridge. The preferred embodiment of the system and Method of the Legacy Data Conversion as described in Appendix 1. Utilizing the invention described in that patent, seamless communication with a legacy host is possible because, in accordance with the incorporated invention, legacy data stored in the Host **150** is converted into XML formatted data that can be used directly by an XML based middleware layer with little transactional overhead.

If the Host **150** is available the requested transaction is processed and the response from the Host **150** is sent, via the middleware bridge, back to the Computer System **200** where the result can be viewed by the transacting agent. The response of the Host **150** is also sent to the Message Center **170** where it can be viewed by the transacting agent at a later time

or by parties other than the transacting agent who have access to the Message Center 170. A transaction cannot be processed either synchronously or asynchronously when the Host 150 is available, but is unable to perform the requested transaction because of problems with the data entered into the computer system by the requesting agent. In this event, the transaction request is sent to a Customer Relationship Management ("CRM") system controller 160 for re-formatting by the requesting agent and the transaction record is moved from its current location in logical memory. In the preferred embodiment of the invention, the transaction record is removed from the Database 130.

FIG. 1, shows the flow of information through the invention when the requested transaction has been submitted to and successfully executed by the host 150. Once this has occurred the transaction record is moved from its current location in logical memory. In the preferred embodiment of the invention the Database is updated to show that the transaction has been successfully completed. Upon successful execution of a transaction, the host's response to the initiating agent's transaction is translated by the middleware and communicated via the Web Portal 110 to the initiating agent during the current web session. For example, if the Host 150 reports that it has successfully completed a transaction involving the alteration of an address field, the success response would be translated into a success message that would then be communicated to the initiating agent (i.e. "You have successfully updated your address"). A transaction identifier could also be supplied in connection with such message in the event that the transaction record is requested.

In the current state of the art, it is normal for transactions to be processed in a purely bi-directional mode – X requests a transaction from host Y which in turn performs the transaction and sends the result to X. If X has ended the web session prior to the transaction being processed the information is commonly only available to X at the next web session on

the Computer System. In many situations, however, person's other than the requesting agent X may need to have access to the results of an Internet transaction. For example, in the medical industry a doctor, patient, nurse or office manager may all need to see the status of a request for insurance coverage sent to the patient's insurer. Moreover, these people may not have access to the requesting agent's computer system. Consequently, in the preferred embodiment of the invention, the Host's response to a transaction request is also sent to a message center. The transaction responses in the message center may be made accessible to anybody with the appropriate access information.

If, as in **FIG. 4**, the Host **150** is not available when the requesting agent submits the transaction request. In this situation, the transaction record is not moved from its current location in logical memory. In the preferred embodiment of the invention the transaction identifier in the Database **130** is not updated.

In a process called polling, the status of all claims is continually being tested. In the preferred embodiment of the invention, the transaction identifiers in the Database are continually tested to determine the status of the requested transactions. One or more of the transaction records that include attributes associated with the unavailable Host **150**, prompt the Transaction Manager **140** to test the availability of each identified host. When, according to the attributes of the transaction records, the Host **150** is considered unavailable, the Transaction Manager **140** periodically monitors the availability of the Host **150**.

Responsive to a test result that increases the probability that the Host **150** is available, the Transaction Manager **140** re-submits a transaction to the identified Host **150**. For example, if a given set of host machines has a regular maintenance schedule, the availability of one host machine in the set would increase the probability that other host machines in the set are also available. Furthermore, as a result of network communication delays (lag), a

reply from a host machine that is available only establishes that the host was available at the time that the test results were transmitted. Finally, the "availability" of a given host may be determined (at least in part) based on the availability of other hosts.

If the identified Host 150 is available and the host successfully executes the transaction, the transaction record is moved from its current location in logical memory. In the preferred embodiment of the invention, once the transaction is executed the Database 130 is updated to show that the transaction has been successfully executed, which may include deletion of the entry from the Database 130. Furthermore, in the preferred embodiment of the invention, if the initiating agent is still online, the host's response to the executed transaction is communicated to the initiating agent during the current web session. However, if the transaction management system determines that the initiating agent is offline, the host's response is communicated to a message center 170, (such as an electronic mail address), which in turn informs the initiating agent of the host's response to the transaction at the beginning of the initiating agent's next web session.

In the preferred embodiment of the invention, a polling process periodically checks the status of all unfinalized transactions. In this way a status report can be created and sent to the Message Center 170 to provide real time feedback to the initiating agent regarding all submitted transactions. The Message Center may comprise any method of displaying the transaction status or the processed transaction, such as a web page where the status/results of requested transactions are posted and updated. The Message Center 170 may be viewed by the requesting agent or any other person provided with access. Alternatively, instead of the Message Center 170, there may be direct communication with the transacting agent, for example, instant messages regarding the status and results of a transaction may be sent to the transacting agent or a wireless communication method may be used to update the transacting

agent regarding the status and results of a transaction request. In a preferred embodiment of the invention, the requesting agent is provided with status and transaction result updates in real-time. In the preferred embodiment of the invention, if the transaction cannot be executed because of a non-recoverable error an error report is generated in order to prompt manual intervention.

**FIG. 3** shows a flow diagram of a preferred embodiment of the system and method when there is synchronous communication between the requesting agent and the required host 150.

The requesting agent submits a transaction, during a web session, via a web portal 110. In the preferred embodiment, the transaction management process is performed in the Enterprise JavaBeans® ("EJB") tier 120, which transforms the transaction into a format that is appropriate for the identified host although any similarly functional alternative could also be used. The Transaction Manager 140 tests the availability of the required Host 150. In this diagram, the Host 150 is available, therefore, the Transaction Manager 140 submits the transaction to the identified Host 150 via a Transport Layer 330. In turn, the Host 150 executes the transaction and sends a reply to the Transaction Manager 140. The Transaction Manager 140 identifies that the transaction has occurred synchronously and sends the hosts reply to the EJB tier 120. The EJB tier 120 translates the Host's reply into a format that is appropriate for the web portal. The host's reply to the transaction is then displayed on the Web Portal 110 for the initiating agent during the web session.

**FIG. 4** shows a flow diagram of a preferred embodiment of the system and method when there is asynchronous communication between the requesting agent and the required host.

The initiating agent submits a transaction, during a web session, via a Web Portal 110. The Enterprise EJB tier 120 transforms the transaction into a format that is appropriate for the identified Host 150. The Transaction Manager 140 tests the availability of the required Host 150. In this diagram the Host 150 is unavailable, therefore, the Transaction Manager 140 queues the transaction and does not update the applicable database. The Transaction Manager 140 periodically tests the availability of the required Host 150. Once the Transaction Manager 140 identifies that the required Host 150 is available it sends the transaction to the identified host via the Transport Layer 330. In turn, the Host 150 executes the transaction and sends a reply, via the Transport Layer 330, to the Transaction Manager 140. The Transaction Manager 140 identifies that the transaction has occurred asynchronously and sends the host's reply to a Message Center 170. Thus, when the initiating agent retrieves messages during a subsequent web session, it will receive host's reply to the transaction.

FIG. 5 shows a flow diagram and side-by-side comparison, of preferred embodiments of the invention, for synchronous and asynchronous communication between the initiating agent and the host.

As discussed previously, a transaction/request or transactions/requests are sent via a web portal to a host. The Transaction Manager, part of the middleware, identifies the required host or hosts and determines whether the identified host is available.

If the Host is available, the transaction/request is prepared for synchronous communication with the host. The prepared transaction/request is then transmitted to the host. The Transaction Manager monitors whether the transaction/request has been successfully executed by the host. If the transaction/request is not executed the transaction/request is prepared again and resubmitted to the host until successfully executed.

Once the transaction/request is executed by the host, the data model is updated and if transaction/request is a part of a batch, the next transaction/request is executed until all of the batch of transactions/requests have been executed.

If the host is unavailable, the transaction/request is prepared for asynchronous communication. The transaction/request is queued until the host becomes available. When the host is available the queued transaction/request is transmitted to the host. The Transaction Manager monitors whether the transaction/request has been successfully executed by the host. If the transaction/request is not executed the transaction/request is prepared again and resubmitted to the host until successfully executed. Once the transaction/request is executed by the host an alert center is notified, which provides feedback to the initiating agent that its transaction/request has been successfully executed. If the transaction/request is a part of a batch all of the batch of transactions/requests are executed and the alert center is informed that the entire batch has been executed and the initiating agent is informed of this result via the alert center.

**FIG. 6** separates the elements of a preferred embodiment of the invention and shows how they are involved in the process and method.

Every time a request is made via a Web Portal **110**, the Transaction Manager **140** determines whether the system is available.

If the system is available, the Transaction Manager **140** prepares the request for a synchronous transaction with the system and immediately submits the request to the system via the Transport layer **330**. If the request is successfully executed the Transaction Manager **140** updates the data model. If the request is not successfully executed, the request is sent to the CRM system controller **160** for modification and reformatting. After the request has been modified by the CRM system controller **160** it is sent back to the Transaction Manager **140**.

The Transaction Manager 140 then resubmits the request to the system repeating the process with the CRM system controller 160, as necessary, until the transaction is executed and the Model Manager 610 updates the data model.

If the system is unavailable the Transaction Manager 140 prepares the request for an asynchronous transaction with the system. The Transaction Manager 140 sends the request to the MQ ("Message Queue") Controller 620 for queuing. The Transaction Manager 140 monitors the system and once it determines that the system is available, the queued request is immediately transmitted to the system via the Transport Layer 330. If the request is successfully executed by the system the Transaction Manager 140 notifies the user. If the request is not successfully executed the Transaction Manager 140 submits the request via the transport layer 330 to the CRM system controller 160, which makes modifications, as necessary, to the request. The re-formatted request is then returned to the Transaction Manager 140, which sends the request via the Transport Layer 330 to the system, repeating the process with the CRM system controller 160 as necessary until the request is executed. Once the request is executed, the Transaction Manager 140 sends a message via the transport layer 330 to the requesting agent.

The foregoing description of the preferred embodiments of the invention has been presented for the purposes of illustration and description. For example, although described with respect to the insurance and healthcare industries, the system and method could be used in other industries or for nonbusiness reasons. Additionally, the claimed system and method should not be limited to the particular embodiments disclosed. The descriptions of the header structures should also not be limited to the embodiments described. For these reasons, this description is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It



is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

## CLAIMS

What is claimed is:

1. A system and method for managing and monitoring, in real-time, interactions between a healthcare provider and a patient comprising the step of:
  - maintaining a virtual Electronic Waiting Room associated with the healthcare provider.
2. The system and method of claim 1, further comprising making the virtual Electronic Waiting Room accessible to the healthcare provider's employees.
3. A system and method for managing and monitoring, in real-time, interactions between a healthcare provider and a patient comprising the steps of:
  - a. storing one or more attributes relating to the patient;
  - b. storing an identification for the patient;
  - c. creating an electronic representation of the patient, which is linked to said Patient's attributes, in the Electronic Waiting Room when the patient's identification is entered into the office Provider's system at the beginning of a Patient's visit.
4. The system and method of claim 2, wherein said Patient's attributes are updated at the end of the Patient visit to include all information regarding the visit.
5. The system and method of claim 2, further comprising adding work items to the Patient's representation in the Electronic Waiting Room as necessary during the visit.
6. The system and method of claim 2, further comprising a Patient portal through which a Patient can, via the Internet, view a summary of the status of his or her transactions with the office Provider and/or an Insurer.

100 – Flow Diagram Showing Process for Creating a Patient Profile

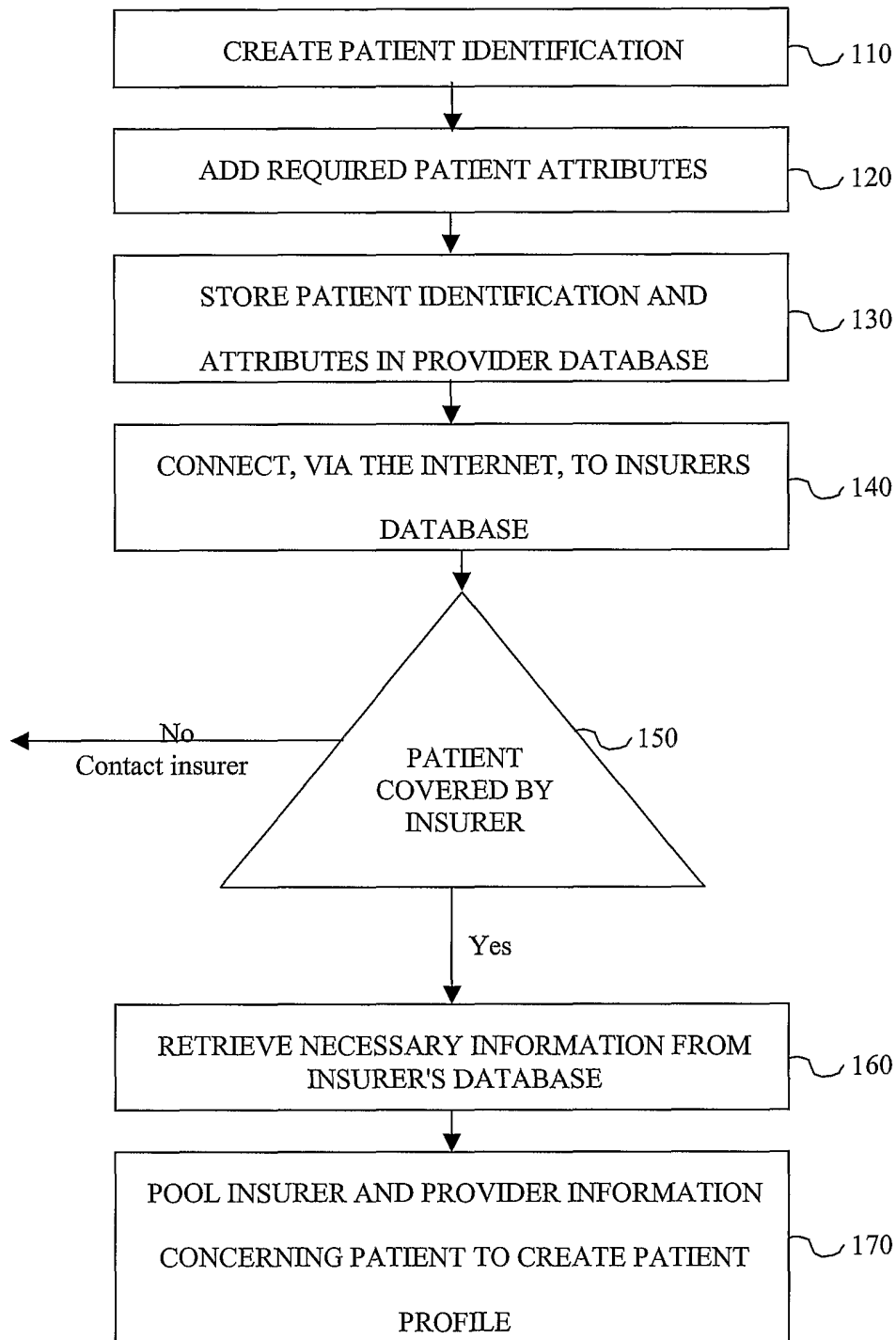


FIG. 1.

200 – Electronic Waiting Room Interface

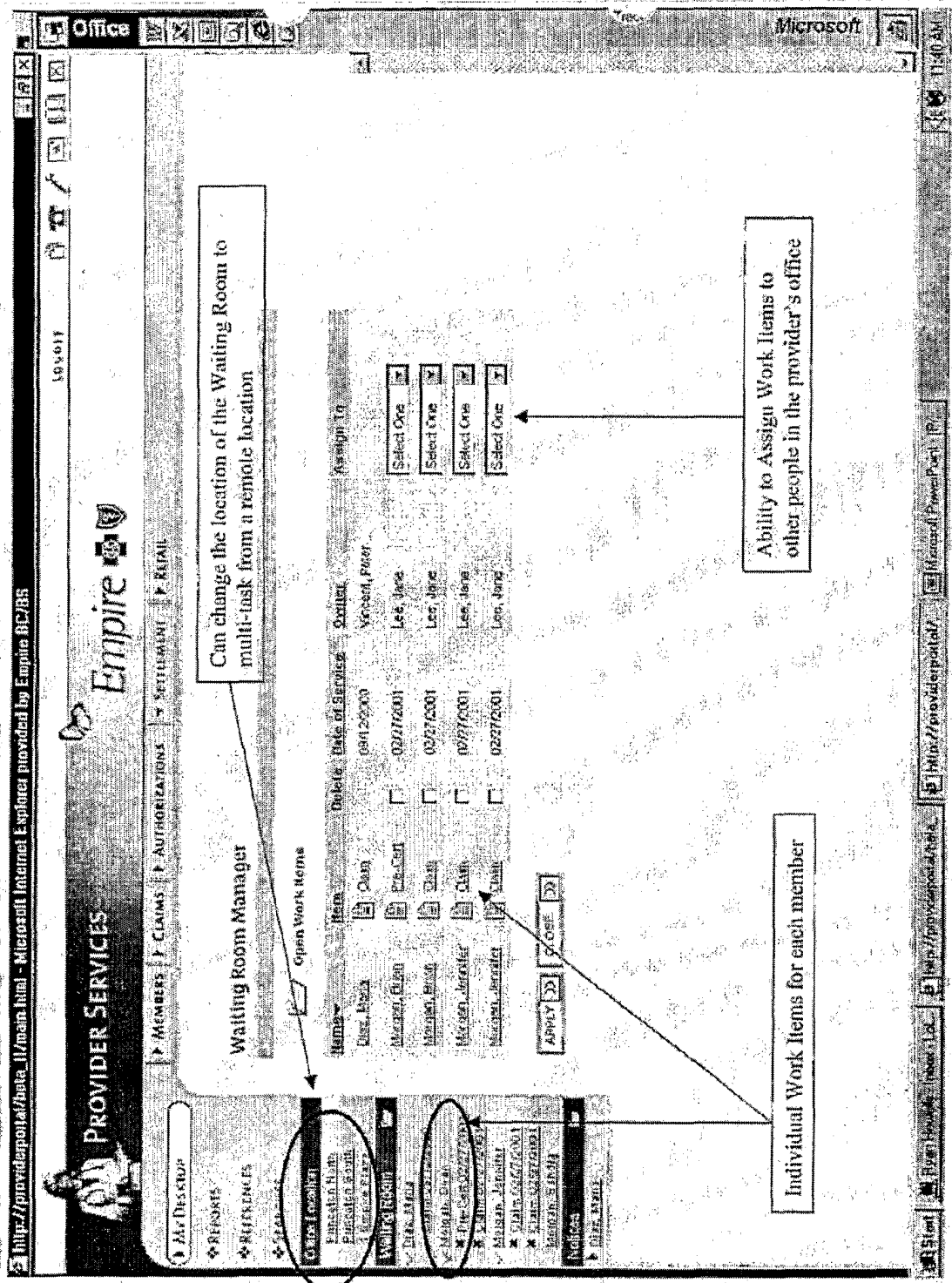


FIG. 2

300 – Flow Diagram Showing a Preferred Embodiment of the Invention

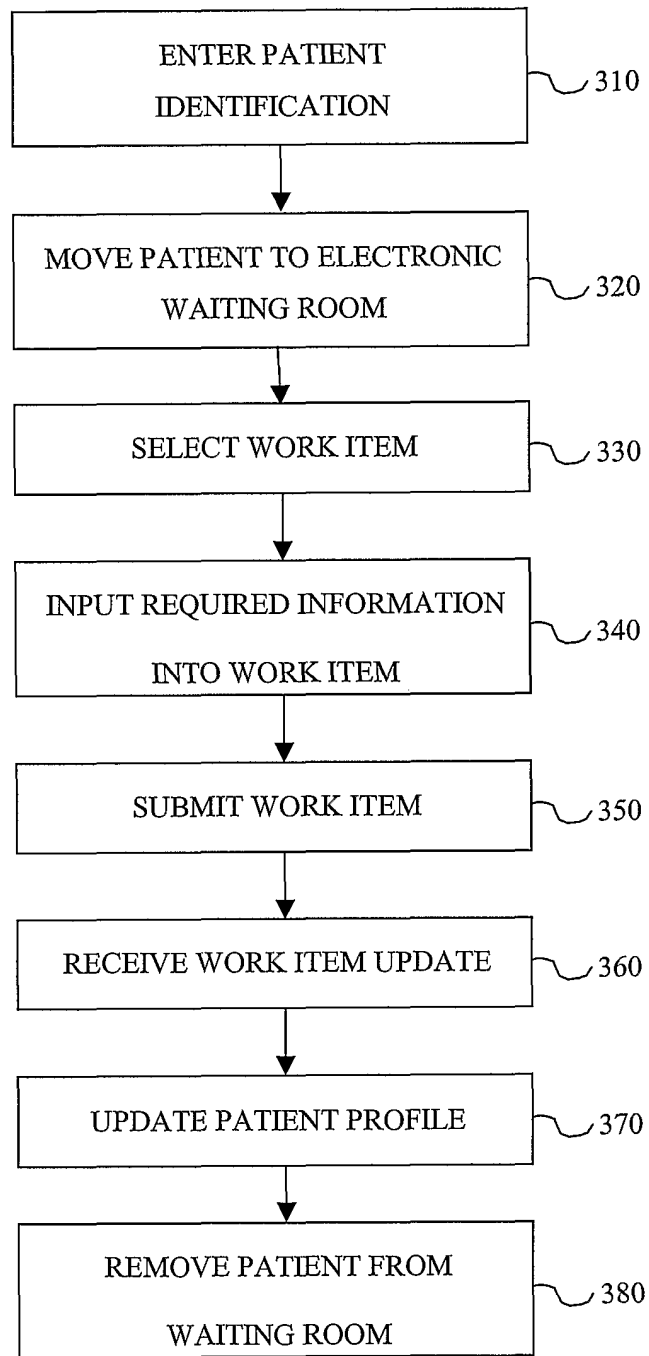


FIGURE 3.

400 – Flow Diagram Showing a Preferred Embodiment of the Invention

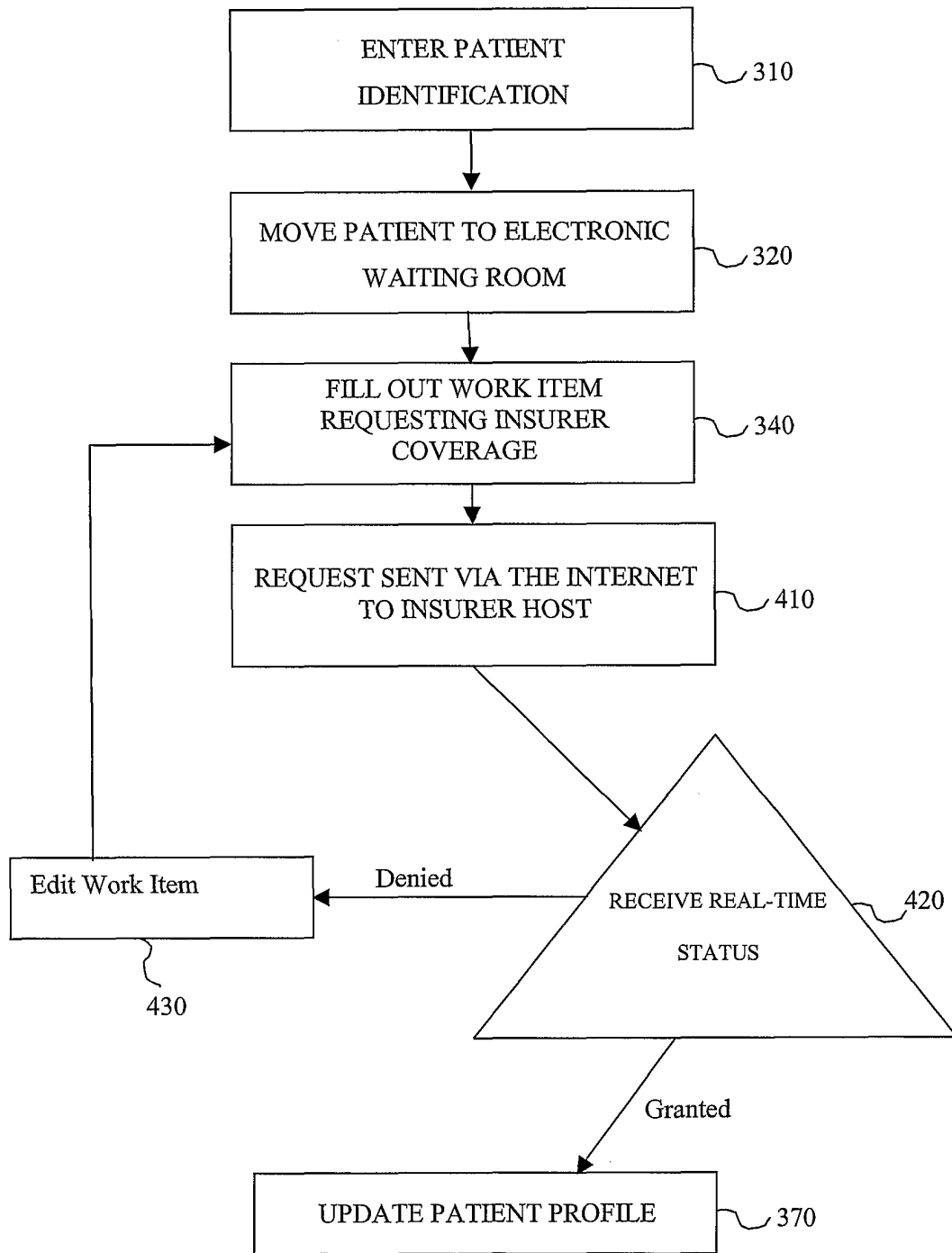


FIGURE 4.

500 -- Electronic Waiting Room Management

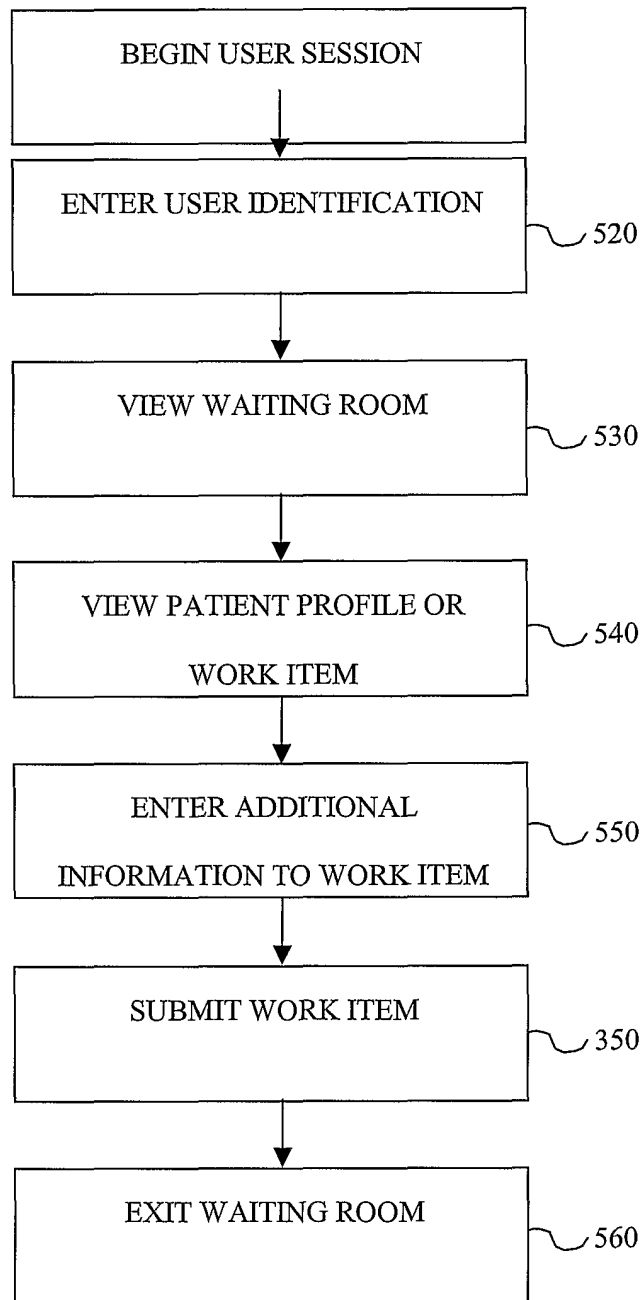


FIGURE 5.

600 – Sharing Electronic Waiting Room Information

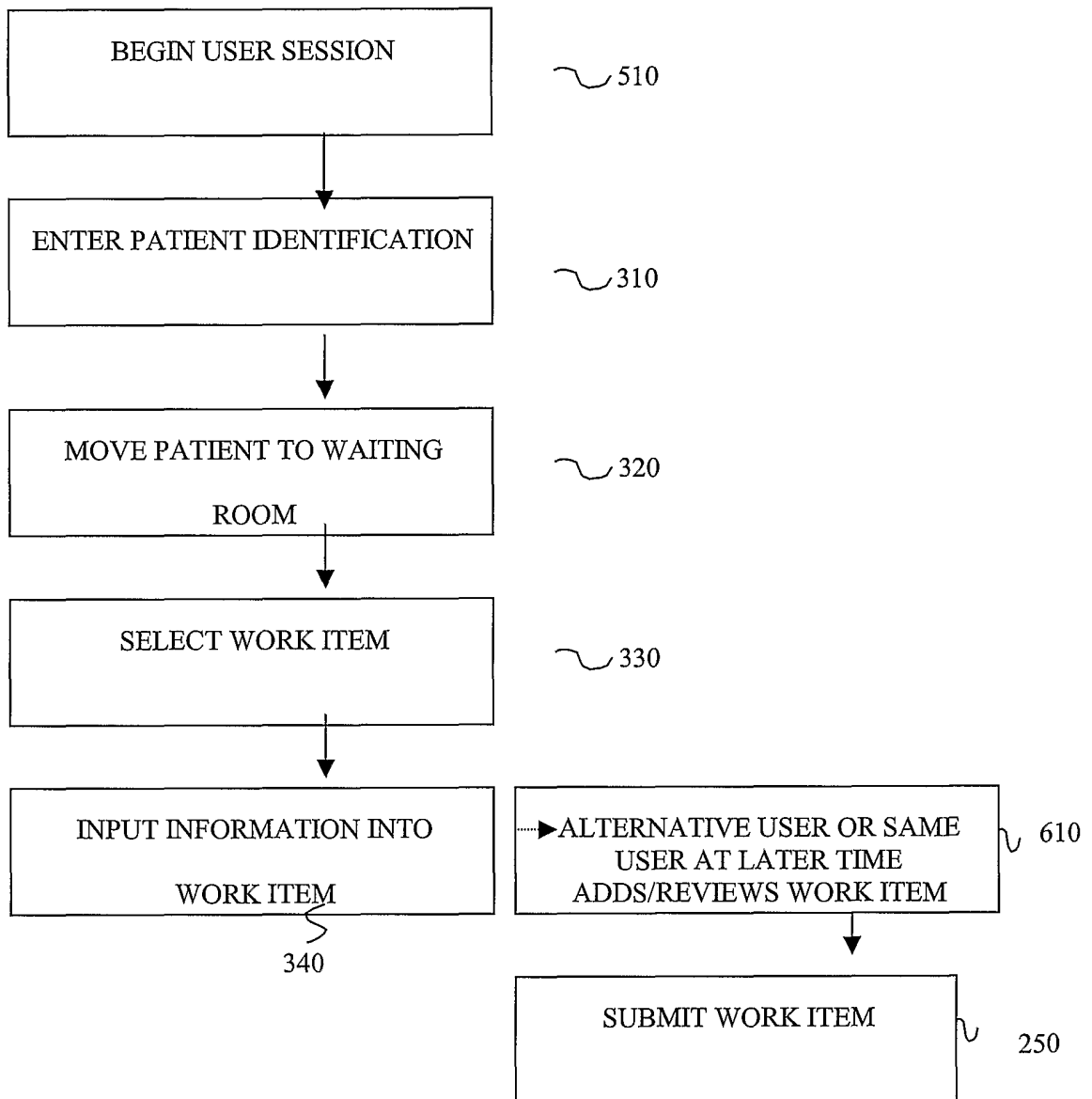


FIGURE 6.



700 – Patient Access to Information

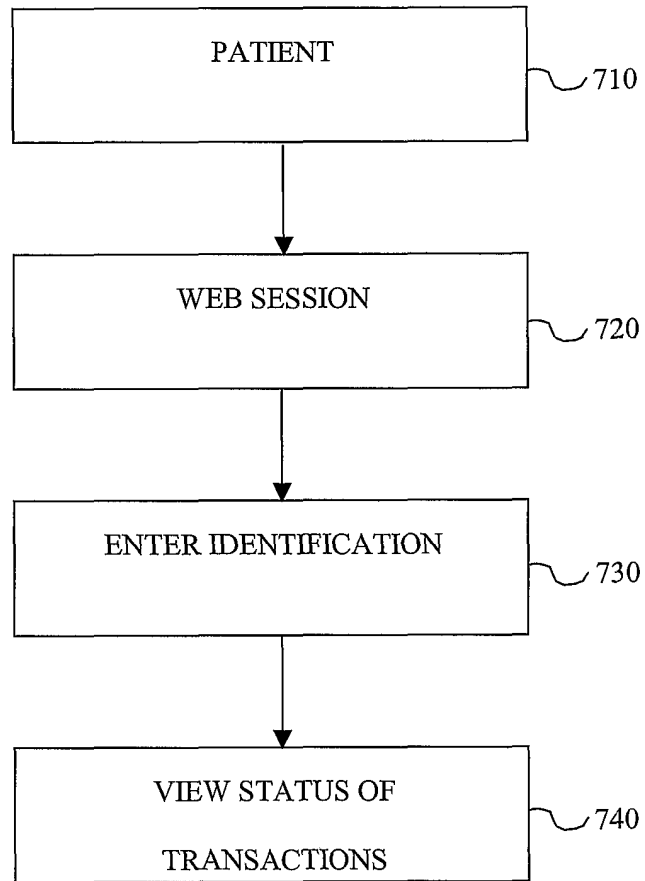


FIGURE 7.