SYSTEM AND METHOD FOR UPLOADING PRODUCT DATA TO A COMMODITY EXCHANGE SERVER

A client/server system and method for uploading product data specifications from a plurality of client computers to an on-line commodity exchange server is provided. The system provides a scalable client/server architecture in which a downloadable interactive uploader applet is transferred to each client computer, which cooperates with an instance of an uploader servlet operating at the server. The downloadable interactive uploader applet is utilized by each client to interactively map the client's product attributes to a set of common product attributes associated with the products capable of being sold via the on-line commodity exchange system. As a client interactively maps their product attributes to the common format, the applet generates an adaptive mapping file for the particular type of product, which is then saved for use in subsequent uploads of the same type of product data so that the client need not repeat the mapping steps. The applet also provides for adaptive data-type mapping in which the client adaptively maps their data types for a particular product attribute to the data types common to the on-line system. The adaptive data-type mapping is also saved in the adaptive mapping files.
System and Method for Uploading Product Data to a Commodity Exchange Server

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention is directed toward the field of electronic commerce (E-commerce). More specifically, the invention provides a system and method for uploading product data to an on-line commodity exchange server (CES), which is then used to manage the buying and selling of the products over an electronic communication network, such as the Internet.

2. Description of the Related Art

Uploading product data specifications from numerous product sellers to a common on-line exchange server is problematic. The problem primarily resides in the fact that each of the product sellers will describe their products with a unique set of product attributes. The issue then becomes how to effectively transform the descriptive product attributes used by each of the sellers into a common set of product attributes that are used by the on-line exchange server.

There are two primary mechanisms that are currently being used to solve this problem: (1) the “integration” method; and (2) the XML method. In the “integration” method, technical personnel associated with the commodity exchange system actually meet and confer with technical personnel from each potential seller in order to manually map the seller’s product attribute fields and data types to the product attribute fields and data types expected by the CES system. This method is highly problematic, however, as it takes a tremendous amount of time and skilled resources to manually map each seller’s data to that expected by the CES system. In addition, if the seller alters their data mapping in any manner, the manual process must be repeated in order to ensure compatibility between the systems.
2

In the XML method, the operators of the CES system utilize the advances brought on by the extensible markup language (XML) to develop a document type definition (DTD) in XML that describes the product attributes and valid data types for the CES system. Then, in order to make seller systems compatible with the DTD, the seller systems must be altered to output an XML datafile organized according to the DTD. This methodology would work if all the potential seller/clients that wish to load product specifications into the CES had sophisticated IT staff personnel that could carry out the proper programming needed to create the XML file. Unfortunately, however, many seller/clients simply do not possess the sophistication necessary to alter their systems to generate such a structured data file. Thus, the XML approach would be limited to only those seller/clients having a high-level of technological sophistication.

SUMMARY OF THE INVENTION

A client/server system and method for uploading product data specifications from a plurality of client computers to an on-line commodity exchange server is provided. The system provides a scalable client/server architecture in which a downloadable interactive uploader applet is transferred to each client computer, which cooperates with an instance of an uploader servelet operating at the server. The downloadable interactive uploader applet is utilized by each client to interactively map the client's product attributes to a set of common product attributes associated with the products capable of being sold via the on-line commodity exchange system. As a client interactively maps their client product attributes to the common format, the applet generates an adaptive mapping file for the particular type of product, which is then saved for use in subsequent uploads of the same type of product data so that the client need not repeat the mapping steps. The applet also provides for adaptive data-type mapping in which the client adaptively maps their data types for a
particular product attribute to the data types common to the on-line system. The adaptive data-type mapping is also saved in the adaptive mapping files.

According to one aspect of the invention, a method is provided for uploading product data from a client system to a server. The method preferably includes the following steps: (1) downloading an interactive product data mapping applet from the server to the client system; (2) executing the applet at the client computer; (3) selecting a transfer file at the client computer that describes the client's products using a set of client product attributes; (4) mapping the client product attributes to a set of server product attributes using the interactive product data mapping applet, and storing corresponding mapping information in an adaptive mapping file; and (5) uploading the transfer file and the adaptive mapping file to the server.

Another aspect of the invention provides a system for uploading product data from a client system to a server, comprising: at least one transfer file at the client computer that describes the client's products using a set of client product attributes; and a downloadable interactive product data mapping applet operating at the client system, wherein the applet is programmed to: (1) interactively map the client product attributes to a set of server product attributes; (2) store corresponding mapping information in an adaptive mapping file; and (3) upload the transfer file and the adaptive mapping file to the server.

Still another aspect of the invention provides a system for uploading product data from a plurality of client systems to a common server. This system preferably includes: an interactive uploader application operating at each of the client systems, the interactive uploader application including software programming codes for interactively mapping client-specific product attributes for each product to be uploaded to server-specific product attributes, and then for storing the interactive mapping information in a mapping file; and an uploader server in communication with the common server for
communicating with the interactive uploader applications and for enabling the transfer of the product data and the mapping file to the common server.

Another, more specific aspect of the invention provides a method of uploading product data from a plurality of client systems to an on-line commodity exchange server via the Internet, comprising the steps of: (1) accessing the on-line commodity exchange server via a web browser operating at the client systems; and (2) for each client system selecting to upload product data to the commodity exchange server operating the steps of: (a) downloading an interactive uploader applet from the commodity exchange server to the client system; (b) executing the interactive uploader applet at the client system; (c) instantiating an instance of an uploader servelet at the commodity exchange server for each executed interactive uploader applet, wherein each uploader servelet is in communication with one interactive uploader applet via a communication channel over the Internet; (d) interactively mapping client-specific product attributes for the product data to be uploaded to server-specific product data attributes and storing associated mapping information into an adaptive mapping file; and (e) transferring the product data and the adaptive mapping file from the client system to the commodity exchange server using the communication channel formed between the interactive uploader applet and the uploader servelet.

These are just a few of the many aspects of the invention, which are detailed below in reference to the attached drawing figure. Other aspects and variations thereof will be apparent to one of ordinary skill in this field upon reviewing this application.

The present invention provides many advantages, such as: (1) provides interactive product data mapping and adaptive learning; (2) provides automatic product data uploading; (3) provides product data validation; (4) provides a server architecture that easily scales to a large number of client users; (5) provides high-speed uploading capabilities; and (6) minimizes the
involvement of trained information technology (IT) personnel in carrying out the product data mapping function.

These are just a few of the many advantages of the present invention, which is described in more detail below in terms of the preferred embodiments. Not all of these advantages are required to practice the invention, and this listing is provided simply to illustrate the numerous advances provided by the invention. As will be appreciated, the invention is capable of other and different embodiments, and its several details are capable of modifications in various respects, all without departing from the spirit of the invention. Accordingly, the drawings and description of the preferred embodiments set forth below are to be regarded as illustrative in nature and not restrictive.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention provides many advantages over presently known systems and methods in this field, as will become apparent from the following description when read in conjunction with the accompanying drawings, wherein:

FIGs. 1A-1B set forth a flow diagram of a preferred methodology for uploading product data to a commodity exchange server according to the present invention; and

FIG. 2 is an exemplary system diagram of the present invention.

These drawing figures present one or more preferred embodiments of the present invention. The preferred embodiments, which are described in detail below, are presented by way of example, and are not meant to limit the scope of the claimed invention.

**DETAILED DESCRIPTION OF THE DRAWINGS**

Turning now to the drawing figures, FIGs. 1A-1B set forth a flow diagram 10 of a preferred methodology for uploading product data to a
commodity exchange server. The process begins at step 12 where a client access the commodity exchange server (CES), preferably over an electronic communication network such as the Internet. As described in more detail below in connection with FIG. 2, the system of the present invention preferably includes a plurality of client computer systems (which could be personal computers, workstations, laptops, or computers connected over a corporate LAN) coupled over the Internet to the CES system.

The CES system is preferably a web site for conducting the on-line buying and selling of a commodity product. In one embodiment of the invention, the CES server is used for buying and selling steel products over the Internet. An exemplary CES system for steel products can be found at http://www.e-steel.com. The functionality of the E-Steel web site is detailed in co-pending provisional patent application serial number 60/152,784 (the '784 application), entitled "Electronic Commodity Exchange System." The teaching of the '784 application is hereby incorporated into this application by reference.

Commodity exchange servers, such as the E-Steel web site, typically provide the ability to buy and sell a variety of commodity products that are defined by a set of product attributes. For example, the E-Steel web site supports a variety of steel-related products such as hot rolled steel, cold rolled steel, coated product, plate steel, tin mill, and rebar, to name a few. Each of these steel products is further defined by a set of product attributes that describe the characteristics of each product type. For example, the rebar steel product type is typically defined by the following product attributes: product class, defect type, quantity, industry standard, standard number, Rockwell B scale, size, length, unit weight, part number, serial number, technical quality, producing mill name, price, packaging information, dimensional tolerance, yield strength, tensile strength, percent elongation, etc. Other product types are defined using similar or other product attributes. As described above, the basic problem that sellers of commodity products encounter when trying to get
their products listed on such a CES system is compatibility between the product attributes that are utilized by the CES system in describing the types of products that can be sold over the system and the product attributes that are used by the individual client sellers to describe their products.

Turning back now to FIG. 1A, the client seller accesses the CES system at step 12. In the preferred embodiment of the invention, this step involves the client engaging a web browser application at their system, and then connecting to the CES web site over the Internet. Once connected to the CES web site, the client then typically inputs a username/password login combination in order to gain authorized access to the CES system. Once the client has logged into the system, at step 14 the client selects a text or graphical display that indicates to the system that the client desires to upload new product data specifications to the CES system.

At step 16, the CES system determines whether the particular client has previously uploaded product data to the system. If not, then at step 18 the CES system downloads an interactive uploader applet program to the client system by transmitting the applet from its storage location in the CES database to the local memory of the client’s computer system. As seen below in FIG. 2, the preferred system of the present invention provides a client/server uploader architecture that includes uploader applet programs on the client computers and a corresponding servelet program operating on the CES system.

The interactive uploader applet program is preferably a Java applet that runs within the context of the client’s web browser application and provides the functionality described in the remainder of FIGs. 1A-1B. The Java applet only needs to be downloaded once to the client’s system, where it is installed locally. Thus, if at step 16 the CES system determined that the particular client had previously uploaded product data to the system, then step 18 can be skipped since the applet is already stored at the client’s computer. Although not shown specifically in FIG. 1A, the CES system could, alternatively, determine whether the client system is operating the most current version of
the interactive uploader applet, and if not, the CES system would then proceed to step 18 and download the most recent version of this application to the client’s system.

Regardless of whether the CES system downloads the interactive uploader applet at step 18 or not, the method then proceeds to step 22, where the CES system determines whether or not this particular client has any adaptive mapping files (AMFs) stored in the CES database 20, and if so, downloads these AMF files to the client system for use with the interactive uploader applet. As described in more detail below, the adaptive mapping file contains information that maps the client’s product attribute fields and data types to the product attribute fields and data types common to the CES system. For each client there can be one or more AMF files stored at the CES database 20, each AMF file corresponding to a particular product type that may be sold through the CES system. These AMF files are created during the interactive mapping process described below, and subsequently uploaded to the CES system along with the mapped and validated product data.

Once the AMF files (if any exist) are downloaded to the client system, the method then proceeds to step 24, where the system displays an informational screen to the client system through the client’s web browser application. As noted above, in the preferred embodiment of the invention, the client interacts with the CES system through a web browser program, such as Microsoft’s Internet Explorer™ or Netscape’s Communicator™. This easy to use, intuitive type of interface makes it relatively simple for the client to interact with the CES system, and to prepare the client’s product data for uploading. The informational screen at step 24 basically describes to the client the subsequent steps listed in the remainder of FIGs. 1A-1B so that the user can understand how the uploader applet operates and what information the client will need to provide in order to get their product data onto the CES system.
At step 26, the client creates a transfer file (TF). The transfer file is preferably either an ASCII delimited data file or an XML file. The transfer file has a tabular structure organized into rows and columns, where each row represents a particular product that the client wants to upload to the CES system, and the columns of the table represent the client’s product attributes associated with that type of product. The transfer file may be created off-line prior to the client accessing the CES system, such that the transfer file is ready to process when the client accesses the system. Having created the transfer file, the client then loads it into the memory space of the applet at step 28 for subsequent processing. This step preferably includes a browse function where the client uses a standard Windows browse dialog box to locate the transfer file on the client’s system and to thereby instruct the applet where the transfer file resides so that the applet can load the transfer file into its memory space.

Having loaded the transfer file into the applet’s memory space, the client then selects the type of product at step 30. This step is important because it tells the applet which CES system product attributes should be associated with the product data in the transfer file. The applet includes a list of required product attributes used by the CES system for each supported product type. In the example of the E-Steel commodity exchange system, the preferred product types are hot rolled, cold rolled, coated, plate, tin mill and rebar. Each of these six products types has their own specific product attributes that describe the product to the buyers in the CES system. Generally speaking, the product attributes that are programmed into the CES system for describing particular product types are different than the product attributes used by the various seller/clients that utilize the system. Thus, a mechanism is required for interactively mapping the client’s product data attributes to the product attributes utilized by the CES system. In the present invention, this mechanism is the interactive uploader applet.

After the applet has loaded the raw transfer file into its memory space and the client has indicated the product type, the applet then checks to see if
the transfer file is larger than a specified maximum file size at step 32. In order to prevent exceedingly long delays when uploading files over a modem connection, as well as to prevent bogging down the CES system, the method of the present invention provides a maximum file size that can be uploaded by the interactive uploader applet. If that maximum file size is exceeded, then control of the method passes to step 34, and an error message is displayed through the client’s web browser indicating that the transfer file is too large.

Assuming the transfer file does not exceed the maximum file size, control of the method then passes to step 36, where the applet determines whether an adaptive mapping file (AMF) exists for the selected product type. If this AMF file existed, it would have been downloaded from the CES system to the client’s computer in step 22. This would then indicate that this client has previously gone through the interactive mapping and data validation steps 44-52 for this same type of product, and thus the applet should apply the mapping information stored in the AMF file to the client’s new transfer file for this same product.

If an AMF file is not found in step 36, then control passes to step 42, where the interactive uploader applet takes the raw data from the transfer file prepared by the client and maps the raw data into a display table for display to the client through the web browser interface. The transfer file raw data is mapped into the display table as a series of rows and columns. Each row in the display table corresponds to a particular product that the client desires to upload to the CES system for subsequent sale. The columns of the display table correspond to the CES system product attributes for this particular type of product. Since there is no AMF file to apply to the transfer file in step 42, the columns are initially labeled “Unassigned,” since the client has not yet mapped their column type to the product attributes utilized by the CES system. Each column in the display table has an associated column number, depending on the number of columns that exist in the transfer file. Thus, the display table
shows the contents of the raw data from the transfer file as a series of rows and columns, with each column having an associated column number.

Because the client previously selected the product type to associate with this particular transfer file, the applet knows what product attributes the CES system needs to associate with this particular product type. Concurrently with displaying the raw data from the transfer file in the display table, the applet also displays a listing of each of the required product attributes for the particular product, and next to that product attribute provides an input box so that the client can indicate which column number in their transfer file is associated with each required product attribute.

At step 44, the client then reviews the required product attributes for this type of product and inputs the column number of the display table that is associated with those product attributes. (The applet 72 preferably checks and validates the column numbers to ensure that the client has input a valid numerical value based on the number of columns in the display table. The applet may also verify that the client has not input two column numbers that have the same value.) This column association information maps the client’s raw transfer file to the product attribute fields required by the CES system for this particular product. After the client has interactively mapped the data columns in the raw transfer file to the product attributes of the CES system, the applet then stores this mapping information into a new AMF file associated with this particular client and this particular product type. The AMF file can then be used in subsequent uploads of the same type of product data so that the client need not re-map the raw transfer file product data to the format required by the CES system.

If, at step 36, the interactive uploader applet determined that an AMF file did exist for this particular product type (indicating that the client had previously gone through the interactive mapping steps), then control passes to step 38, where the applet displays the same display table discussed above in step 42, but now it applies the mapping information stored in the AMF file.
Thus, instead of the columns being labeled as "Unassigned," where column association information exists in the AMF file, this information is used by the applet to label the column with the appropriate product attribute. These product attribute associations were previously mapped by the client when the AMF file was first created by the client for this particular product.

In this manner, the display table adapts based on the prior interactive mapping associations selected by the client, and stored in the AMF file. At step 40, the client is given the opportunity to review the mapping associations applied from the AMF file and to change those associations if they are no longer valid. If the associations are still correct, then control passes to step 48. If, however, the prior associations stored in the AMF file are no longer valid, then control passes back to steps 44 and 46 (described above), and the client can then interactively map the invalid columns into the correct product attributes required by the CES system. This new mapping information is then re-saved into the AMF file in step 46 for subsequent uploads of the same product type.

The ability to provide this AMF file enables the possibility of providing clients with a "one-click" uploading function for their product data. In this one-click upload, assuming that the product attribute associations have not changed, the client can simply specify the product type (as in step 30) and then select "upload." The interactive uploader applet would then apply the information stored in the AMF file, validate the data in the file, and then transmit the transfer file to the CES system. Without the adaptive learning provide though the AMF file, this type of one-click upload process would not be feasible.

From either of steps 40 or 46, control of the preferred methodology passes to step 48, in which the applet reviews the display table (and its contents) for exceptions. There are two primary types of data exceptions that should be addressed by the applet. The first is the situation where the CES system requires a particular product attribute that is not reflected in the client's transfer file. The second is the situation where the content of the client's
transfer file includes data that is not valid a valid data type for the associated product attribute.

In the first situation, where the client’s transfer file does not include data associated with a required product attribute of the system, the applet appends blank columns to the display table and labels those blank columns with the missing product attributes. Having modified the display table in this manner, the client can then manually input the required data items into the blank columns.

In the second situation, the applet will highlight any client data from the transfer file that is not valid based on the product attribute selected by the client for that particular column of data. For example, and returning to the E-Steel system, the client may have associated the product attribute “Product Class” with a particular column of data. In the E-Steel system, the valid data entries for the product attribute “Product Class” may be “Prime” and “Non-Prime.” But the particular client refers to “Non-Prime” product as “NP.” Thus, the client’s data is not a valid data type for this particular product attribute. The methodology then passes to step 50, where the client maps any invalid data to valid data types for the particular product attributes selected. This is typically done by first highlighting the invalid data item in the display table, and then providing the client with a drop-down window or selection box which includes only the valid data entries. So, for example, next to the highlighted invalid entry “NP,” the applet may display a selection window with the valid data selections “Prime” and “Non-Prime.” The client would then select the “Non-Prime” designation, which would then result in two actions taking place. First, the applet would scan the raw data from the transfer file and replace any instance of “NP” with “Non-Prime,” and second the interactive data mapping information that indicates that “NP” equals “Non-Prime” for this particular client and this particular product would be stored in the AMF file for use in processing subsequent uploads of the same type of product data.
Having mapped the columns to the correct product attributes and resolved any data exceptions, the applet then proceeds to step 52, where it validates the data loaded into the display table. If there are any remaining invalid or out-of-range data items, control reverts to step 50 until these invalid data items are put into a form that is acceptable to the CES system.

Following data validation, the applet then sends a message to the CES system indicating that a file transfer is being requested. The CES system acknowledges that it is ready to proceed with a file transfer, and the applet then transmits the mapped and validated data in the transfer file to the CES system, where the data is stored in an area of the CES database 20 referred to as the “holding bin.” The holding bin is an area of the CES system where client’s can individually store and manage their product data and subsequently make those products openly available for sale through the CES system. The applet then transmits the AMF file(s) to the system for storage in the CES database 20 and for subsequent use during additional upload sessions.

During the file transmission process, the applet collects transmission statistics on the amount of data being transferred, and the time of transfer. This information may be displayed through the web browser interface to the client so that the client can monitor the uploading process. The uploading process ends at 58.

As noted above, the first time that a particular client uploads a particular product type, the client must go through the interactive mapping process for the product attribute columns and for any invalid data entries. This information is saved in the AMF files, transferred to the CES system, and subsequently downloaded to the client’s system to streamline the uploading of subsequent product data.

FIG. 2 is an exemplary system diagram of the present invention. The system 60 preferably includes a primary commodity exchange server (CES) system 62 and a plurality of client systems 64, wherein the plurality of client systems 64 are coupled to the CES system 62 via an on-line communication
network, such as the Internet 66. It should be understood, however, that although a single CES system 62 is shown in FIG. 2, this system could be a distributed system including a number of different servers or computers, which may be coupled via a local area network (LAN) or via a wide area network (WAN). It should also be understood that although a single client system 64 is shown in FIG. 2, there could be a plurality of client computers coupled via a LAN or WAN, all of which may be coupled to the CES system 62 via the Internet 66.

The client system 64 is preferably a PC or a workstation. This system 64 preferably includes a web browser application, such as Internet Explorer or Communicator, operating in conjunction with a transmission control protocol and internet protocol (TCP/IP) interface 68, which enables bidirectional communication with other computers via the Internet 66. The TCP/IP interface 68 could be a dial-up adapter interface such that the client computer 64 is coupled to the Internet 66 via a dial-up modem connection, or the interface 68 could be a LAN interface, in which the client system 64 is communicating information back and forth over a LAN infrastructure, which is in turn coupled to an Internet firewall computer that separates the client’s protected LAN from the unprotected public Internet 66.

The client system also includes the Java interactive uploader applet 72 (which enables the functionality described above in FIGs. 1A-1B), one or more transfer files 76, and one or more adaptive mapping files (AMFs) 74. As described above, the interactive uploader applet 72 is stored on the CES system 62 within the CES database 20, and is downloaded to the client’s system 64 when the client first makes a request to upload product data to the CES system 62. The applet may be re-downloaded if it is updated and the client does not have the most current version of the applet. In either case, the applet is stored within the local storage of the client’s system 64. The transfer file(s) 74 are stored locally on the client’s system (or may be located on some network drive in the LAN embodiment), and the adaptive mapping files 74 are
temporarily stored in the local storage of the client's system 64 each time the client engages the uploading process depicted in FIGs. 1A-1B.

The server-side system 62 provides a scalable servelet architecture that couples to and communicates with the plurality of applets 72 operating at the client systems 64. Together, the servelet and applet software components of the overall system provide an applet/servelet architecture that can be scaled to serve a multitude of clients who may be simultaneously uploading large transfer files of product data to the server 62.

The server 62 communicates via the Internet 66 using a TCP/IP interface module 68. This interface 68 is coupled to a server 78, which is preferably a Unix server hosting the CES web site 80, and is also operating a Java virtual machine 82. The CES web site 80 is preferably a commodity exchange web site where buyers and sellers can come together to effect electronic transactions for the specific commodity being sold though the system. An example of such a CES web site system is the E-Steel web site for buying and selling steel products. This web site is described in more detail in the '784 application, which has been incorporated into this application by reference. The Java virtual machine 82 provides an interface between the web site server 80 and the Java uploader server 86.

The scalability of the CES system 62 is provided by including a Java uploader server 86, which is preferably implemented on a separate Unix server than the web server 80 and the Java virtual machine 82. The Java uploader server 86 receives requests for uploading product data from the web server 80, and in response to each request, the Java server 86 instantiates an instance of the Java uploader servelet 88. Thus, for each upload operation engaged to the plurality of client systems 64, there is a separate instance of the Java uploader servelet 88. This provides the scalability advantage of the present architecture.

Each Java uploader servelet 88 communicates with a corresponding Java uploader applet 72 operating on one of the client systems 72. It is this servelet 88, for example, that determines whether this particular client 64 has
previously uploaded product data to the CES system 62 (as shown in step 16 of FIG. 1A), and then, in response to this determination step, the servelet 88 controls the downloading of the interactive Java uploader applet 72 to the client system 64. The servelet 88 also determines whether this client 64 has any stored AMF files 74 at the CES system 62, and if so, communicates these AMF files 74 over the Internet 66 to the client’s system 64 so that the mapping information can be adaptively applied to the client’s transfer file 76. The server 88 also communicates with the applet 72 to effect the transfer of the mapped and validated transfer file from the client system 64 to the CES database 20. The CES database 20 stores the current version of the Java uploader applet 72, the AMF files 74 for each client, and also provides the holding bin area 90 where the client’s mapped and validated transfer files are stored.

When a particular client system 64 desires to upload a new transfer file 76 to the CES system 62, it sends a request over the Internet 66 to the CES web server 80. The web server 80 receives the request and opens a secure socket connection 84 to the Java uploader server 86 via the Java virtual machine 82. The Java uploader server 86 then instantiates an instance of the Java uploader servelet 88 for the particular request. The servelet then transfers the Java uploader applet 72 and any adaptive mapping files 74 to the client system 64, if necessary. The client then utilizes the applet 72 to process, map and validate the data in the transfer file 76 as described above in FIGs. 1A-1B. Having processed the transfer file, the client system 64 then sends an upload request to the servelet 88.

The servelet 88 then communicates back to the applet 72 and together these components manage the transfer of the mapped and validated data over the Internet 66 to the CES system 62. Once received by the servelet 88, the transfer file is further processed by parsing the file, validating the data in the file, and converting the file to a business object that is compatible with the CES system 62. This business object transfer file is then stored in the holding
bin 90 area of the CES database 20. Subsequently, the client can enter the holding bin area of the CES web site 80 and make the products stored in the transfer file available for sale through the CES system.

The preferred embodiments described with reference to the attached drawing figures are presented only to demonstrate certain examples of the invention. Other elements, steps, methods and techniques that are insubstantially different from those described above and/or in the appended claims are also intended to be within the scope of the invention.
What is claimed:

1. A method of uploading product data from a client system to a server, comprising the steps of:
   downloading an interactive product data mapping applet from the server to the client system;
   executing the applet at the client computer;
   selecting a transfer file at the client computer that describes the client’s products using a set of client product attributes;
   mapping the client product attributes to a set of server product attributes using the interactive product data mapping applet, and storing corresponding mapping information in an adaptive mapping file; and
   uploading the transfer file and the adaptive mapping file to the server.

2. The method of claim 1, further comprising the steps of:
   prior to downloading the interactive product data mapping applet, determining at the server whether the client has previously uploaded product data to the server; and
   if the client has previously uploaded product data to the server, then only re-downloading the interactive product data mapping applet to the client computer if there is a new version of the applet.

3. The method of claim 1, further comprising the step of:
   loading the transfer file into the memory space of the interactive product data mapping applet.
4. The method of claim 1, further comprising the steps of:
   selecting the product type of the product data to be uploaded from the
   client system; and
   based on the product type selected, determining which server product
   attributes to use in the mapping step.

5. The method of claim 1, further comprising the steps of:
   determining whether the transfer file exceeds a predetermined
   maximum file size; and
   if the transfer file exceeds the predetermined maximum file size, then
   terminating the method prior to uploading the product data and displaying an
   error message at the client system.

6. The method of claim 1, further comprising the steps of:
   determining whether the client system has been utilized previously to
   upload product data to the server; and
   if the client system has been so utilized, then applying a stored
   adaptive mapping file to the client’s product data prior to the mapping step,
   wherein the adaptive mapping file includes previously stored information that
   maps the client’s product attributes to the server product attributes.

7. The method of claim 1, further comprising the step of:
   validating data types contained in the client’s transfer file with a set of
   server data types associated with the server product attributes.

8. The method of claim 1, further comprising the steps of:
   if the validating step indicates that certain data types contained in the
   client’s transfer file do not match the server data types associated with the
   server product attributes, then:
mapping the client's data types to the set of server data types
using the interactive product data mapping applet;
and storing corresponding mapping information in the adaptive
mapping file.

9. The method of claim 1, further comprising the steps of:
   displaying transmission statistics as the transfer file is uploaded to the server.

10. The method of claim 1, further comprising the steps of:
    downloading the adaptive mapping file from the server to the client system;
    selecting a second transfer file at the client computer that describes the client's products using a set of client product attributes;
    applying the mapping information stored in the adaptive mapping file to the data in the second transfer file; and
    uploading the second transfer file to the server.

11. The method of claim 1, further comprising the steps of:
    providing a plurality of product types, each product type having an associated set of server product attributes that describe the product type;
    repeating the mapping step at least once for each client and each product type that the client uploads to the server; and
    storing the associated adaptive mapping files generated in the mapping step for each product type uploaded by a particular client system.

12. A system for uploading product data from a client system to a server, comprising:
means for downloading an interactive product data mapping applet from the server to the client system;
means for executing the applet at the client computer;
means for selecting a transfer file at the client computer that describes the client's products using a set of client product attributes;
means for mapping the client product attributes to a set of server product attributes using the interactive product data mapping applet, and storing corresponding mapping information in an adaptive mapping file; and
means for uploading the transfer file and the adaptive mapping file to the server.

13. A system for uploading product data from a client system to a server, comprising:
at least one transfer file at the client computer that describes the client's products using a set of client product attributes; and
a downloadable interactive product data mapping applet operating at the client system, wherein the applet is programmed to: (1) interactively map the client product attributes to a set of server product attributes; (2) store corresponding mapping information in an adaptive mapping file; and (3) upload the transfer file and the adaptive mapping file to the server.

14. A system for uploading product data from a plurality of client systems to a common server, comprising:
an interactive uploader application operating at each of the client systems, the interactive uploader application including software programming codes for interactively mapping client-specific product attributes for each product to be uploaded to server-specific product attributes, and then for storing the interactive mapping information in a mapping file; and
an uploader server in communication with the common server for communicating with the interactive uploader applications and for enabling the transfer of the product data and the mapping file to the common server.

15. The system of claim 14, wherein the interactive uploader application is a Java application.

16. The system of claim 14, wherein the product data is organized into a transfer file at the client systems, the transfer file being defined by a plurality of rows and columns, wherein each row is associated with a particular product to be uploaded to the common server, and wherein each column represents the product attributes for that product.

17. The system of claim 14, wherein the interactive uploader applet further includes software programming codes for: (a) downloading mapping files from the common server for products that were previously uploaded to the common server by the particular client system; and (b) automatically mapping the client-specific product attributes to the server-specific product attributes using the information in the mapping files.

18. The system of claim 14, wherein the common server further comprises: a TCP/IP interface for communicating over the Internet; a commodity exchange web server interface for enabling the buying and selling of commodity products; and a Java virtual machine coupled between the commodity exchange web server and the uploader server for enabling the secure communication of product data between client systems and the uploader server through the commodity exchange web server.
19. The system of claim 18, wherein the common server further comprises:
   a database for storing: (a) the interactive uploader application; (b) the
   mapping files; and (c) the uploaded product data.

20. The system of claim 14, wherein the uploader server instantiates an
   instance of an uploader servelet for each interactive uploader applet operating
   at the client systems, thereby forming an applet/servelet pair for each client
   attempting to upload product data to the server.

21. A method of uploading product data from a plurality of client systems to
   an on-line commodity exchange server via the Internet, comprising the steps
   of:

   accessing the on-line commodity exchange server via a web browser
   operating at the client systems;

   for each client system selecting to upload product data to the
   commodity exchange server operating the steps of:

   (a) downloading an interactive uploader applet from the
       commodity exchange server to the client system;

   (b) executing the interactive uploader applet at the client
       system;

   (c) instantiating an instance of an uploader servelet at the
       commodity exchange server for each executed interactive uploader
       applet, wherein each uploader servelet is in communication with one
       interactive uploader applet via a communication channel over the
       Internet;

   (d) interactively mapping client-specific product attributes for
       the product data to be uploaded to server-specific product data
       attributes and storing associated mapping information into an adaptive
       mapping file; and
(e) transferring the product data and the adaptive mapping file from the client system to the commodity exchange server using the communication channel formed between the interactive uploader applet and the uploader servelet.

22. The method of claim 21, wherein the commodity exchange server is used for buying and selling steel products over the Internet.
1/3

CLIENT ACCESSES COMMODITY EXCHANGE SERVER (CES)

SELECT UPLOAD PRODUCT DATA TO CES

HAS CLIENT PREVIOUSLY UPLOADED PRODUCT DATA?

YES

DOWNLOAD INTERACTIVE UPLOADER APPLET TO CLIENT SYSTEM

DOWNLOAD ADAPTIVE MAPPING FILE(S) TO CLIENT SYSTEM

DISPLAY INFORMATION ON OPERATION OF INTERACTIVE UPLOADER APPLET

CREATE A TRANSFER FILE (TF) FOR PRODUCT DATA

LOAD THE TRANSFER FILE INTO APPLET MEMORY SPACE

SELECT PRODUCT TYPE

IS TF > MAXIMUM SIZE?

YES

DISPLAY ERROR MESSAGE AND RETURN

NO
A

NO

ADAPTIVE MAP FILE EXISTS FOR THIS PRODUCT?

DISPLAY RAW ROW/COLUMN DATA AND UNMAPPED PRODUCT TYPE ATTRIBUTE FIELDS

MAP COLUMNS TO PRODUCT TYPE ATTRIBUTE FIELDS

STORE MAPPING INFORMATION INTO ADAPTIVE MAPPING FILE

REVIEW DATA EXCEPTIONS: (1) ENTER MISSING DATA FOR COLUMN EXCEPTIONS; (2) VALIDATE DATA IN ATTRIBUTE FIELDS

MAP INVALID DATA TO VALID DATA ATTRIBUTES AND STORE IN ADAPTIVE MAPPING FILE

YES

CHANGE MAPPING?

NO

ALL VALID DATA?

YES

TRANSMIT MAPPED/VALIDATED DATA AND AMF TO CES AND STORE

DISPLAY TRANSMISSION STATISTICS

END

Fig. 1B