Title: SYSTEM AND METHOD FOR INTERFACING A DATA PROCESSING SYSTEM TO A BUSINESS-TO-BUSINESS INTEGRATION SYSTEM

Abstract: In accordance with the invention, a method for interfacing a data processing system (e.g., 12) to a business-to-business integration system (18) uses an event-based triggering technique to automatically engage in an electronic transaction between different business entities to minimize errors with human intervention in the transaction process. The creation of a transaction is sensed by detecting the occurrence of a triggering event (S12). Reference data is read for the transaction from an application database upon the detection of the triggering event (S14). An outgoing message is formed from transactional data and the reference data upon the detection of the triggering event (S15).
SYSTEM AND METHOD FOR INTERFACING A DATA PROCESSING SYSTEM TO A BUSINESS-TO-BUSINESS INTEGRATION SYSTEM

FIELD OF THE INVENTION

This invention relates to a system and method for interfacing a data processing system to a business-to-business integration system to facilitate the transfer of data for a commercial transaction between different business entities.

BACKGROUND

A data processing system refers to an enterprise resource planning system (ERP) or any computer system that supports the operation of a business entity. An enterprise resource planning system is a data processing system that supports one or more operational functions of a business entity. An enterprise resource planning system may facilitate the integration of business functions, such as materials management, planning, purchasing, and accounting for a business entity.

The data processing system may be designed primarily to handle the internal operations of a single business entity without conducting electronic communications with a remote data processing system of another business entity. If a business entity wishes to conduct a transaction with another business entity via electronic communications, a business-to-business integration system may be used in the communications path that links the data processing systems of the respective business entities. However, because of limitations of the interface between the data processing system and the business-to-business integration system, the execution of transactions between the business entities may not occur in real-time and the execution may require human intervention.

Real-time refers to a computational process that is executed within a maximum time period after the occurrence of an event, where the time period is preferably imperceptible to a human user of the computational process. The user may perceive the execution of the computational process through a user interface (e.g., graphical user interface) coupled to the data processing system. If transactions do not occur in real-time, then the transactional data available to the business entities does not represent current data. As a result, computations based
on the outdated transactional data may contribute to an error in a subsequent transaction. The cumulative effects of the multiple delays over many transactions may lead to material inefficiencies in the operations of the business entities and deficient financial performance, because future transactions and business decisions may be based on inaccurate or outdated data.

Human intervention may be required to trigger the formation of a transaction, the execution of a transaction, to track the status of a transaction, or to audit the transaction between two business entities. Human intervention entails labor costs. Further, if the execution of a transaction depends upon the worker's entry of data via a user interface, typographical errors or other clerical errors may cause the execution of erroneous or inaccurate transactions.

In the past, several standard techniques have been used for integrating an enterprise resource planning (ERP) system and a business-to-business integration system. Under a first integration technique, a conventional remote function call (RFC) communication is used for communications between the enterprise resource planning system and the business-to-business integration system. Human intervention is generally required to enter or initiate a transaction in a manner consistent with conventional operation of the enterprise resource planning system, for example. Further, the conventional RFC does not provide a tracking or an auditing mechanism for monitoring the status of a transaction or the accuracy of historic transactions.

Under a second integration technique, the business-to-business integration system may include a converter to convert a standard application document (e.g., ERP document) from the enterprise resource planning system into a standard exchangeable format document or an I-DOC communication. An I-DOC communication refers to an intelligent document (e.g., a document based on portable document format (PDF), hypertext mark-up language (HTML), or JPEG for images) that is derived from an application document (e.g., ERP document).

The second technique suffers from several infirmities owing to the I-DOC communication. First, an I-DOC communication does not typically support real-time or stream-based communications because of the extensive processing
resources required to translate and convert the format of the application document to an I-DOC communication. Stream-based communications refer to communications that are continuous, once established, such that large blocks of information may be efficiently transferred over a virtual or actual communications link. Second, the I-DOC communication does not adequately support synchronous execution of multiple transactions. Synchronous execution refers to execution of a group of transactions in sequential order, one immediately following another. The data processing delay associated with converting each successive application-based document into a corresponding I-DOC communication tends to prevent the synchronous execution of the group, unless the synchronous execution is initiated after waiting for the protracted conversion of the entire group. Third, the I-DOC communications does not sufficiently support control of various business models and requirements in a modular fashion.

Thus, a need exists for an improved electronic interface or software interface between a data processing system a business-to-business integration system that supports real-time transactions and eliminates or reduces human intervention in the facilitation of transactions.

SUMMARY OF THE INVENTION

In accordance with the invention, a system and method for interfacing a data processing system to a business-to-business integration system uses an event-based triggering technique to automatically engage in an electronic transaction between different business entities to minimize errors with human intervention in the transactional process. The creation of a transaction is sensed by detecting the occurrence of a triggering event. Reference data is read for the transaction from an application database upon the detecting of the triggering event. An outgoing data message is formed from transactional data and the reference data upon the detecting of the triggering event. The system and method of the invention is well suited for supporting real-time transactions between data processing systems of different business entities with transaction auditing capability and minimal human intervention for transaction execution.
BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of communications infrastructure that includes a communications interface for interfacing a data processing system to a business-to-business integration system in accordance with the invention.

FIG. 2 is a block diagram that shows the communications interface of the data processing system of FIG. 1 in greater detail.

FIG. 3 is a flow chart of method for interfacing a data processing system to a business-to-business integration system in accordance with the invention.

FIG. 4 is a block diagram of an alternate embodiment of communications infrastructure that supports interfacing a data processing system with a business-to-business integration system in accordance with the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

In accordance with the invention FIG. 1, shows communications infrastructure 11 that includes a first data processing system 12 coupled to a business-to-business integration system 18. In turn, the business-to-business integration system 18 communicates with the second processing system 24 via a communications network 22 (e.g., Internet). In one embodiment, the first data processing system 12 and the business-to-business integration system 18 are associated with a first business entity, whereas the second data processing system 24 is associated with a second business entity.

The first data processing system 12 includes a first business application 14 and a communications interface 16. Further, the communications interface 16 may be in communication with an output database 34 for storing transactional data. The first business application 14 supports the electronic management of a business or operational function of at least the first business entity. An operational function may include, but is not limited to, accounting, materials management, maintenance-repair organization, planning, purchasing, engineering, shipping, logistics, electronic-commerce, and management. In one embodiment, the first business application 14 comprises an SAP ERP program. SAP software is a
trademark of SAP America, Inc., which is affiliated with SAP systems, Applications and Products in Data Processing of Germany.

The first data processing system 12 and the first business application 14 may support modular expansion or modification of operational functions. The first business application 14 may include a group of software modules for conducting a transaction between a first business entity and a second business entity. Each software module is composed of various components. One transaction may use a first subset of the components within the group, whereas another transaction may use a second subset of the components for another transaction. To the extent different transactions share the same software components, the design and modification of business functions is simplified by shorter development times, and frequently shorter program codes to accomplish a given objective.

The communications interface 16 of the first data processing system 12 supports an event-based triggering mechanism. The triggering mechanism triggers the initiation of an outgoing data message for a business transaction between the first business entity and the second business entity. In one embodiment, the outgoing data message is based on an ERP document outputted from an output port 13 of the first data processing system 12. The outgoing data message refers to a data message that originates at the first data processing system 12 and traverses the communications network 22 to the second data processing system 24. An incoming data message originates at the second processing system 24 and traverses the communications network 22 to the business-to-business integration system 18.

The communications interface 16 promotes the exchange of outgoing messages between the first data processing system 12 and business-to-business integration system 18 in real-time. Accordingly, a transaction between the first data processing system 12 and the second data processing system 24 may be executed in real-time. Real-time refers to a computational process (e.g., transaction) that is executed within a maximum time period after the occurrence of an event, where the time period is preferably imperceptible to human user of the
computational process via a user interface (e.g., user interface 10). Real-time may refer to the ability of the first data processing system 12 and the first business application 14 to respond to an event within the maximum time period after the occurrence of the event.

The first data processing system 12 may operate in a synchronous event-driven mode or an asynchronous event-driven mode. An event-driven mode means that upon detection of a triggering event, in response, the first data processing system 12 invokes an appropriate software module of the first business application 14 or the communications interface 16. In a synchronous event-driven mode, the first data processing system 12 services an entire series of events prior to returning control of the first data processing system 12 to a user interface 10. Thus, the user may notice a delay because of waiting for return of control of the user interface 10. The entire series of events may require processing to complete or initiate a transaction between the first and the second business entities.

In an asynchronous event-driven mode, the first data processing system 12 services events on an individual basis rather than a batch basis. Thus, the asynchronous event-driven mode intersperses time segments for responsive processing to events. A user interface 10 interacts in a manner such that the user does not notice delay from the asynchronous event processing. Multiple time segments for asynchronous event processing are typically required to support the initiation or completion of a transaction.

The communications interface 16 provides software instructions to facilitate communications between the first data processing system 12 and the business-to-business integration system 18 via an output port 13 of the first data processing system 12. In general, the output port 13 may comprise a transceiver or another interface that is suitable for or intended for communications with a printer, a facsimile modem, a modem device, a network device, a transceiver, or another electronic device. In one example, the output port 13 represents a universal asynchronous receiver-transmitter (UART) that supports asynchronous serial communications with a modem. In another example, the output port 13 represents a parallel port that supports unidirectional or bi-directional transfer of
data to an electronic device. In one embodiment, the communications interface 16 cooperates with the request handler 50 to make the conversion process more efficient than it would otherwise be by filtering an outputted ERP document.

In the context of SAP, remote function calls (RFC's) are program instructions that are used to configure the output port 13 for communication with an external device. An RFC may use output message determination to send an outgoing data message from the communications interface 16 to the business-to-business integration system 18.

The business-to-business integration system 18 includes a business-to-business integration program 20. The business-to-business integration system 18 may represent a server that runs a suitable business-to-business integration program 20 that satisfies the functional requirements (i.e., data format conversion, communications monitoring and data message authentication) set forth herein. For example, WebMethods B2B software may satisfy at least some of the functional requirements set forth herein. WebMethods B2B software is a trademark of webMethods, Inc. The business-to-business integration system 18 includes a communications manager 80 as a component of the business-to-business integration program 20. The communications manager 80 may comprise a request handler 50 and an authenticator 84. In one embodiment, the request handler 50 includes a converter and a communications monitor. The converter translates or converts an initial data format (e.g., a filtered ERP document) into an appropriate data format (e.g., XML document or another format compatible with hypertext transmission protocol (HTTP)) for an outgoing communication over the communications network 22. The communications monitor monitors and enhances the reliability of communications between the first data processing system 12 and the business-to-business integration system 18.

The business-to-business integration system 18 or the communications manager 80 includes an authenticator 84 for validating the authenticity of a data message transmitted from the second data processing system 24 to restrict access of unauthorized users to the business-to-business integration system 18 and the first data processing system 12.
In an alternate embodiment, the business-to-business integration system 18 further includes a security module for limiting access of authorized users to the first business application 14 of the first data processing system 12 and preventing unauthorized users for accessing the first business application 14 altogether. A security system may comprise a firewall or a filter for filtering data packets based on header content, for example.

The business-to-business integration system 18 may send extensible mark-up language (XML) documents to the second data processing system 24 in data packets or another suitable manner over the communications network 22. Extensible mark-up language (XML) provides guidelines or rules for structuring data in an object-oriented, hierarchical textual file. XML supports the Secure Socket Layer (SSL) for secure communications and public-key encryption of data. XML is compatible with HyperText Transfer Protocol (HTTP) and HyperText Transfer Protocol, Secure (HTTPS). HTTP defines the format and the transmission parameters of data messages over a communications network 22 (e.g., Internet).

The second data processing system 24 includes an interpreter 86. The interpreter 86 interprets outgoing data messages that are sent by the first data processing system 12 via the business-to-business integration system 18 and the communications network 22. The outgoing data messages may refer to XML or HTTP messages represented by data packets or otherwise. The second data processing system 24 may include a second business application 26 that addresses one or more of the following operational functions for at least the second business entity: (1) supply management, (2) shipping logistics, (3) purchasing, (4) e-commerce, (5) accounting applications, and (6) planning and inventory management applications. The second data processing system 24 may be programmed to handle the same operational functions as the first data processing system 12.

In one example, the first data processing system 12 supports an SAP ERP program as the first business application 14. An SAP ERP program typically has various modules, including sales and distribution and materials management. The
communications output of the sales and distribution module is referred to as output
determination. The communications output of the materials management is
referred to as message determination, although in practice the technical parameters
of the output determination are generally equivalent to the technical parameters of
the message determination.

In general, the first business application 14 (e.g., SAP ERP program)
creates a transaction that is treated as a triggering event. The formation or creation
of a transaction may represent a triggering event. Such formation of a transaction
may occur manually with human intervention or automatically incidental to the
operation of the first data processing system 12. In accordance with one example,
the materials management module of the first business application 14 may create a
transaction, a transactional element, or a purchase order as a triggering event. In
accordance with another example, the sales and distribution module of the first
business application 14 may create a transaction, a transactional element, a sales
order, or a factory order, as a triggering event. A transactional element represents
a component or term of a transaction. A sales order or factory order refers to an
internal order of a supplier (e.g., first or second business entity). A sales order or
factory order may be based upon a purchase order issued by a customer of the
supplier.

In accordance with one configuration, an optional transaction auditing
system 88 may be coupled to the first data processing system 12 or elsewhere.
The auditing system 88 is shown in dashed lines because the auditing system is
optional. The auditing system 88 may be supported by referencing an output
database 34 (e.g., NAST table of an SAP ERP system in the output database) for
the first business application 14. The output database 34 may contain
transactional data that is dynamically updated on an on-going basis as transactions
occur or are initiated.

The auditing system 88 refers to a computer or database management
system programmed with software for querying and analyzing the output database
34 (e.g., NAST table). The NAST table refers to a standard table in a database
associated with an SAP ERP system. Any time the first data processing system 12
detects or responds to an event, the detection or executed response may be logged into a table in the output database 34 (e.g., the NAST table). Accordingly, the table in the output database 34 can track every time a message trigger 32 occurs to provide an audit trail. The audit trail may be used to maintain accurate books and records and comply with internal control procedures for monitoring the financial aspects of the electronic transactions between the first business entity and the second business entity. The audit trail may also be extended to other trading partners other than transactions between the first business entity and the second business entity.

FIG. 2 is a block diagram of the communications infrastructure of FIG. 1, where the communications interface 16 is shown in greater detail. Like reference numbers in FIG. 1 and FIG. 2 indicate like elements. The communications interface 16 includes software that may be installed on the first data processing system 12 or a separate computer coupled thereto. FIG. 2 illustrates constituent components of the software of the communications interface 16. The lines interconnecting the various components of the communications interface 16 and the first data processing system 12 represent physical data paths, logical data paths, or both.

The communications interface 16 includes a message trigger 32 that communicates with a driver 42. The driver 42 is responsive to a triggering event detected by the message trigger 32. The first business application 14 may create a transaction; such a transaction may represent a triggering event.

The first business application 14 (e.g., an SAP ERP program) may make an ERP document (e.g., an SAP document), representative of a transaction accessible, to the communications interface 16 as a triggering event. The ERP document may include functional code plus a table of transactional data. A driver 42 (e.g., an Advanced Business Application Programming (ABAP)) of the communications interface 16 is responsive to the triggering event. The occurrence of a triggering event may be indicated by receipt of the ERP document from the first business application 14.
The driver 42 initiates control of the transmission of the ERP document to the business-to-business system 18 upon the occurrence of the triggering event. For example, the driver 42 may follow a priority scheme in the transmission of an ERP documents or a data message. In one embodiment, the driver 42 may follow instructions of an SAP-customized code consistent with Advanced Business Application Programming (ABAP). The computer code may be customized through an SAP developer to manage the processing of outgoing data messages associated with one or more transactions.

The filter 92 of the communications interface 16 filters the ERP document to remove functional code from the ERP document to facilitate enhanced conversion in real-time by the request handler 50. For proper operation and reliable transactions based on up-to-date data, a business-to-business integration system 18 may require the exclusion of the functional code in the ERP document (e.g., SAP document). The functional code refers to the code or data processing instructions that is executable by the first business application 14 (e.g., SAP ERP program) or any similar business program. The functional code may detract from the maximum throughput that would otherwise be available between the first data processing system 12 and the business-to-business integration system 18.

In one embodiment, the filter 92 comprises a Remote Function Call (RFC) that places an ERP document into a suitable format for sending to the business-to-business integration system 18 by filtering out functional code. A Remote Function Call (RFC) provides a mechanism for the first business application 14 or the communications interface 16 to communicate with an external device. For example, the RFC may be used to allow the SAP ERP to send one or more outgoing data messages or filtered ERP documents on the transaction to the business-to-business integration system 18.

In accordance with an alternate embodiment, a business-to-business integration system 18 may actually retrieve the outgoing data message from the communications interface 16, rather than sending outgoing data messages from the first data processing system 12 to the business-to-business integration system 18. The first data processing system 12 may provide a buffer memory (affiliated with
the sender 48) for storing outgoing data messages of filtered ERP documents for such retrieval by the business-to-business integration system 18.

The first business application 14 includes a transaction creator 28 that communicates with a message trigger 32 and an application database 36. In turn, the message trigger 32 communicates with an output database 34 and a driver 42. The application database 36 includes reference data 38 for defining characteristics of the transaction. The driver 42 includes a data reader 44, a queue manager 46, and a data message sender 48.

Upon detection of a created transaction, the message trigger 32 triggers the formation of one or more data messages or an ERP document representative of the transaction or at least one transactional element. The created transaction may be expressed as a textual file that contains transactional data (e.g., ERP document). The data reader 44 extracts or reads relevant reference data from the application database 36 to form an outgoing data message. The outgoing data message or ERP document is formed based on the transactional data and the reference data that is affiliated with the transactional data for a particular transaction. Reference data may apply to multiple transactions and may represent background data about the first business entity, the second business entity, general shipment terms, trading partner preferences, payment terms, contractual terms, or the like. In contrast, transactional data may refer to the terms of a particular transaction. Transactional data may vary from transaction to transaction. Transactional data includes terms of a transaction, such as price, delivery date, quantity of a good, an identity of a good, transaction-specific shipment terms, or the like.

After or upon triggering the driver 42, the message trigger 32 may track each outgoing data message or transaction by recording a corresponding entry in the output database 34. The transactions may be stored as one or more tables in the output database 34. The output database 34 provides historical transactional data and reference data for the transactions. The auditing system 88 may access the historical transactional data. The auditing system 88 assures the integrity of the processes used to conduct one or more transactions between the first business entity and the second business entity.
The data messages or ERP documents are stored in the transaction queue 40 for transmission in conformity with a priority scheme. The sender 48 or ERP documents receives data messages from the transaction queue 40. The data message sender 48 communicates with a request handler 50 or listener of the business-to-business integration system 18. A filter 92 may be associated with the data message sender 48 as shown in FIG. 2 so that outgoing data messages or ERP documents to the business-to-business integration system 18 are filtered. The request handler 50 is associated with a business-to-business integration program 20.

The operation of the communications interface 16 may be better understood by considering the outbound communications from the first data processing system 12 to the business-to-business integration system 18 separately from the inbound communications from the business-to-business integration system 18 to the first data processing system 12.

The communications interface 16 uses a message trigger 32 to eliminate at least some degree of human intervention in the exchange of information between business trading partners, such as the first business entity and the second business entity. The message trigger 32 responds to (1) an occurrence of an electronic event, (2) an electronic hand-off from an electronic event or (2) a human entry into a user interface 10 to invoke the driver 42. The occurrence of an electronic event includes the automated formation of a transaction or saving of a file representative of the transaction by the first business application 14. A human entry includes a user's entering a save command into a user interface 10. A save command invokes a software component for execution that saves the file in a storage device or memory component associated with the first data processing system 12. The save command may be manually invoked or automatically invoked by the first business application 14 as the triggering event for the message trigger 32.

The driver 42 is preferably event-driven such that the saving of a purchase order or the formation of a transaction by the first business application 14 (e.g., ERP system) represents a triggering event. The triggering event triggers the first data processing system 12 to start the outgoing message transmission process by
invoking the driver 42. Other examples of triggering events may include sending of the purchase order to a printer (not shown) coupled to the first data processing system 12 (e.g., via output port 13), the expiration of a timer on a regular basis (e.g., daily or weekly) after accumulating at least one data message or ERP document.

The driver 42 controls the invocation of the reader 44, the queue manager 46, and the sender 48. In addition, the driver 42 may complete business calculations and do data manipulation. The driver 42 calls a reader 44 to read or to extract reference data 38 from the application database 36 that may form at least a portion of the outgoing data message or ERP document. The reader 44 may access and retrieve reference data 38 from one or more tables (e.g., SAP tables) of the application database 36. The reference data 38 along with entered or detected transactional data 30 forms the basis of an outgoing data message or outgoing ERP document. In the context of a purchase order, for example, the transactional data 30 may include a material identifier, a plant location, a price, and a quantity for a transaction. Once a read function is completed, the reader 44 provides the reference data 38, the transactional data 30, or both for an outgoing data message or outgoing ERP document to the driver 42.

The driver 42 supports the transmission of the data message or ERP document from the first data processing system 12 to the business-to-business integration system 18. The driver 42 establishes an appropriate data protocol, including error handling, for the outgoing data message or ERP driver. The driver 42 may organize the data message into data fields consistent with a data structure. For example, the driver 42 may assign a transaction identifier to the data message to distinguish the data message from other messages and allow tracking of the performance of the subsequent transmission of the data message.

The sender 48 receives the outgoing data message from the driver 42. The sender 48 may execute a portal setup scheme in which an intermediate data structure is mapped into an output data structure that is compatible with and readable by the business-to-business integration system 18. The output data structure conforms to a shell or framework in which data fields are organized.
In the context of an SAP software configuration, the sender 48 may use the RFC output message determination to establish and fill the shell of the output data structure. The sender 48 includes a filter 92 that filters out ERP-executable or intelligible code (e.g., SAP executable code) from the intermediate data structure to yield the output data structure. For example, the sender 48 may map selected data fields of the intermediate data structure, excluding executable code, to the output data structure. In the context of an SAP software configuration, the sender 48 calls the driver 42 to execute the portal setup scheme and transmission procedure, instead of merely calling up a peripheral device coupled to the output port.

The ERP-executable code may be incompatible with the business-to-business integration system 18 or may represent superfluous code that is not suitable for execution by the business-to-business integration system 18. In the context of an SAP program supported by the first data processing system 12, the executable code may represent programmable logic (e.g., ABAP logic) that is defined in accordance with a developer. If the executable logic or code is excluded from the output data structure, the output data structure may be referred to as a blank data structure. The output data may be organized as a table for output by the sender 48. The sender 48 may packetize the table or send the table as a series of data blocks.

The communications interface 16 maps or filters the executable code from the outgoing data message to allow an ERP system to effectively interact with a business-to-business integration system 18. The communications interface 16 automates communications to reduce or minimize human intervention in transactions between the first business entity and the second business entity to diminish clerical entry errors and associated inaccurate transactions. The event-driven triggering of the sending of an outgoing data message provide an interfacing technique that supports real-time transactions and synchronous execution of a group of transactions in quick succession to each other.

In a preferred embodiment, the communications interface 16 leverages off any availability of event-driven features of the first business application 14, such
as an SAP ERP system. For example, the communications interface 16 may
extend the functionality of the message trigger 32 to trigger software responses
outside of the first business application 14 to foster automated interaction with a
business-to-business integration system 18.

The business-to-business integration system 18 accepts the outgoing data
message or ERP document in the output data format. A request handler 50 of the
business-to-business integration system 18 provides an acknowledgement message
as feedback to the first data processing system 12, if the outgoing data message
was successfully received at the business-to-business integration system 18. If an
acknowledgement is not received within some defined time after the transmission,
the driver 42 may log a communications error and authorize the retransmission of
the outgoing data message by the sender 48. The communications error may be
selected from a list of possible communication errors.

The outgoing message data from the business-to-business integration
system 18 to the second data processing system 24 may be expressed in the form
of an extensible mark-up language, a hypertext mark-up language, or another
suitable data structure. The request handler 50 may convert the ERP document or
outgoing data message from the sender 48 into an extensible mark-up language
file or another suitable data structure. The outgoing data message may contain one
or more transactions. Each transaction preferably includes transaction data and
reference data. A transaction may include the generation of a purchase order to
buy a transactional subject (e.g., good or service). Extensible mark-up language is
advantageous because the language is considered generally platform-independent
and promotes compatibility of informational exchanges between the first data
processing system 12 and the second data processing system 24.

FIG. 3 illustrates a method for interfacing an enterprise resource planning
system (e.g., SAP system) to a business-to-business integration system 18 (e.g.,

In step S10, a first data processing system 12 creates a transaction, which
may be expressed as a transactional document. For example, the data processing
system creates a purchase order defined as the transactional document. The
transaction may be created in response to user input at a user interface 10 of the
data processing system or by a transactional creator 28 in response to an
automated procedure for creating transactions based on the satisfaction of defined
conditions. The transactional document may be an ERP document (e.g., an SAP
document), an object-oriented textual file, or another data structure. In one
embodiment, the transactional document contains transactional data 30 on the
transaction and is organized in accordance with an intermediate data structure.

In step S12, a message trigger 32 detects the creation of a transaction by
detecting the occurrence of a triggering event. For example, the first data
processing system 12 (e.g., ERP component thereof) may save the transactional
document as a file upon formation of the transaction. Saving a file or ERP
document is an action of the first data processing system 12 that may be defined as
a triggering event, although in other embodiments other triggering events may be
defined. The file may be saved automatically in memory or a storage device
associated with the first data processing system 12 without human intervention or
in response to the entry of commands into a user interface 10. Upon detection of
the triggering event, the message trigger 32 invokes a driver 42 in real-time to
form and process an outgoing data message or ERP document. Accordingly, the
driver 42 may process the transaction associated with the outgoing data message in
a real-time manner, as opposed to a delayed manner with human intervention that
might otherwise compromise the integrity or accuracy of a series of inter-
dependent transactions.

In one embodiment, step S12 may use a modification of an existing event-
based output management scheme of an enterprise resource planning program or
system (e.g., an SAP system) associated with the first data processing system 12.
The modification involves modifying the existing event-based output management
scheme to invoke the driver 42 and executing driver instructions that support the
reader 44, the queue manager 46, and the sender 48.

In unique mode in accordance with the invention, the modification of the
existing event-based management scheme (i.e., the modified event-based
management scheme) includes the message trigger 32 and the driver 42. The
message trigger 32 detects the triggering event and, in response, invokes driver instructions of the driver 42 for the formation and transmission of the outgoing data message or ERP document to the business-to-business integration system 18. The driver instructions send the outgoing data message or ERP document to the business-to-business integration system 18 via the output port 13, rather than sending data to an output device coupled to the first data processing system 12 without invoking the driver instructions of the driver 42. Thus, the message trigger 32 and the driver 42 cooperate to automatically execute or initiate at least one transaction to avoid delay, typographical errors, clerical mistakes, or other errors associated with human workers that might otherwise conduct transactions.

In step S14, the driver 42 invokes a reader 44 to read or extract reference data 38 for the transaction from an application database 36 upon the detection of the triggering event. The extracted or read reference data 38 provides information for forming an outgoing data message or ERP document representative of the transaction. The extraction of the reference data promotes the formation of a transaction with complete and unambiguous terms, although an entire transaction may be expressed in one or more outgoing data messages.

In step S15, an outgoing data message is formed from a transactional data 30 plus the extracted or read reference data 38 of step S14. The formed outgoing data message is representative of part of the transaction or the entire transaction. In an alternate embodiment, the transaction may be based upon the transactional data 30 or the reference data 38, rather than both.

In step S16, at least a header or characteristic data of the outgoing data message is stored in the transaction queue 40 in accordance with a priority scheme. For example, the communications interface 16 may store outgoing data messages or ERP documents in the transaction queue 40 according to a priority scheme. The data message is placed in and retrieved from the transaction queue 40 in accordance with the priority scheme. The priority scheme may refer to a first-in, first-out (FIFO) transaction queue 40 where an earlier formed outgoing data message is placed into the transaction queue 40 for transmission prior to a subsequently formed outgoing data message.
In step S18, the driver 42 invokes the sender 48 to send the outgoing data message in accordance with the priority scheme. The sender 48 may include filtering and mapping the outgoing data message from an intermediate data format to an output data format, where the output data format is compatible with a business-to-business integration system 18 (e.g., webMethods B2B server). For example, the intermediate data format may represent an ERP document (e.g., SAP document) or another file format. The ERP document may include executable code that is intelligible to the first data processing system 12. The mapping facilitates the filtering of executable program code that might otherwise be disruptive to the operation of the business-to-business integration system 18. Further, even if the executable program code is not disruptive to the business-to-business integration system 18, the executable program code may represent superfluous data to the business-to-business processing system 18, which once eliminated enhances the opportunity for achieving real-time transactions with the exchange of up-to-date data between the first business entity and the second business entity. The elimination or reduction of the superfluous code by filtering reduces the transmission handling capacity of the output port 13 to support a desired data flow or throughput of outgoing data messages between the first data processing system 12 and the business-to-business communications system 18. In turn, the elimination of the executable code promotes the enhanced throughput of XML files or outgoing data messages over the communications network 22.

The sender 48 sends the outgoing data message in the output data format to the business-to-business integration system 18. At the business-to-business integration system 18, a listener or request handler 50 regularly, periodically, or continuously looks for one or more outgoing data messages transmitted to the business-to-business integration system 18 or otherwise made accessible to the business-to-business integration system 18. The business-to-business integration system 18 may convert a proprietary communications standard (e.g., an SAP document), associated with the enterprise resource planning system, to an extensible mark-up language (XML) output file or another suitable file format for outgoing data messages. In an alternate embodiment, the business-to-business
integration system 18 may retrieve outgoing data messages from the first data processing system 12 when the first data processing system 12 releases them or otherwise makes them available.

The communications interface 16 is arranged to communicate with a transaction queue 40. The transaction queue 40 accumulates different outgoing data messages in preparation for transmission of the data messages in accordance with a defined priority scheme. The priority scheme may be a first-in, first-out (FIFO) priority scheme in which an earlier outgoing message for an earlier transaction is transmitted, logged into the transaction queue 40, prior to a later outgoing message for a later transaction. Although filter 92 is integrated with the sender 48 as shown in FIG. 2, in an alternate embodiment, the filter 92 may be located upstream to filter data messages or ERP documents prior to prioritization or storage in the transaction queue 40.

The business-to-business server 54 of FIG. 4 is similar to the business-to-business integration system 18 of FIG. 1, except the business-to-business server 54 of FIG. 4 preferably acts as a proxy server when granting or providing access of the second ERP system 56 to the first ERP system 52. In contrast, the business-to-business integration system 18 of FIG. 1 may or may not act as a proxy server. The business-to-business integration system 18 may translate a transaction expressed in a standard such as an ERP document (e.g., SAP document) to an outgoing data message that has a data structure (e.g., XML) compatible with transmission over the communications network 22. Like reference numerals in FIG. 1, FIG. 2, and FIG. 4 indicate like elements.

The block diagram of FIG. 4 includes a first ERP system 52 coupled to a business-to-business server 54. In turn, the business-to-business server 54 communicates with another business-to-business server 54 over a communications network 22 (e.g., the Internet). A business-to-business server 54 is coupled to a second ERP system 56. Both the first ERP system 52 and the second ERP system 56 include the first business application 14, the application database 36, and the communications interface 16, as previously described herein.
The method and system for interfacing a data processing system supports real-time execution of one or more transactions, stream-based communication of transactions, and synchronous execution of transactions because of automatic initiation and execution of transactions without the need or processing burden of conversion to an I-DOC communication. Further, the method and system features a message trigger 32 that is arranged to invoke driver instructions for real-time execution of a transaction without human intervention, rather than to merely send data to an output device (e.g., a printer coupled to the first data processing system 12). The first business application 14 has program instructions that are preferably arranged in a modular manner to support flexible control and modification of software components of the first business application 14. The auditing system 88 accesses an output database 34 for tracking and monitoring transactions executed in accordance with the procedures of the invention to maintain the integrity of the procedures and transactions to foster a strong e-commerce business relationship between the first business entity, the second business entity, and any other trading partners.

The foregoing description of the system and system describes several illustrative examples of the invention. Modifications, alternative arrangements, and variations of these illustrative examples are possible and may fall within the scope of the invention. Accordingly, the following claims should be accorded the reasonably broadest interpretation, which is consistent with the specification disclosed herein and not unduly limited by aspects of the preferred embodiments disclosed herein.
The following is claimed:

1. A method for interfacing a data processing system to a business-to-business integration system, the method comprising the steps of:
   detecting a creation of a transaction by detecting the occurrence of a triggering event;
   reading reference data for the transaction from an application database upon the detecting of the triggering event; and
   forming an outgoing data message from transactional data and the reference data upon the detecting of the triggering event.

2. The method of claim 1 wherein the detecting step comprises using an event-based output management scheme of an enterprise resource planning system, associated with the data processing system, to detect the triggering event and, in response, to invoke driver instructions for the formation of the outgoing data message.

3. The method of claim 2 wherein the event-based output management scheme is modified to invoke the driver instructions for sending the outgoing data message to the business-to-business integration system via an output port, rather than sending data to an output device coupled to the data processing system without invoking the driver instructions.

4. The method of claim 1 further comprising the step of invoking driver instructions for sending the outgoing data message to the business-to-business integration system consistent with a priority scheme.

5. The method according to claim 1 wherein the detecting step comprises detecting saving a file as the triggering event.

6. The method of claim 1 wherein the detecting step comprises detecting the saving of an output document of the data processing system as the triggering event.
7. The method according to claim 1 wherein the reading step comprises extracting data from a table in the application database.

8. The method according to claim 1 wherein the forming step further comprises converting an intermediate data format of the outgoing data message into an output data format having a data structure compatible with reception by the business-to-business integration system.

9. The method according to claim 1 wherein the forming step further comprises filtering an intermediate data format of the outgoing data message into an output data format by mapping data elements compatible with the business-to-business integration system into the output data format.

10. The method according to claim 1 wherein the forming step further comprises removing executable code from an intermediate format of the outgoing data message to yield an output data format devoid of the executable code to facilitate transfer of the outgoing data message to the business-to-business integration system.

11. The method according to claim 1 further comprising the steps of:
   storing at least a header of the outgoing data message in a transaction queue in accordance with a priority scheme; and
   sending the outgoing data message in accordance with the priority scheme.

12. A system for interfacing a data processing system to a business-to-business integration system, the system comprising:
   a message trigger for detecting the creation of a transaction by detecting the occurrence of a triggering event;
   a data reader for reading reference data for the transaction from an application database upon the detecting of the triggering event; and
a communications interface for forming an outgoing data message from transactional data and the reference data upon the detecting of the triggering event.

13. The system according to claim 12 wherein the message trigger uses an event-based output management scheme of an enterprise resource planning system, associated with the data processing system, to detect the triggering event and, in response, to invoke driver instructions for the formation of the outgoing data message.

14. The system according to claim 13 wherein the event-based output management scheme is adapted to invoke the driver instructions for sending the outgoing data message to the business-to-business integration system via an output port, rather than sending data to an output device coupled to the data processing system without invoking the driver instructions.

15. The system according to claim 12 wherein the message trigger is adapted to detect saving a file as the triggering event.

16. The system according to claim 12 wherein the message trigger is adapted to detect saving an output document from an enterprise resource planning system as the triggering event.

17. The system according to claim 12 wherein the reader is arranged to extract data from a table in the application database.

18. The system according to claim 12 wherein the communications interface includes a sender for converting an intermediate data format of the outgoing data message into an output data format having a data structure compatible with the business-to-business integration system.

19. The system according to claim 12 wherein the communications interface includes a sender for filtering an intermediate data format of the outgoing
data message into an output data format by mapping data elements compatible with the business-to-business integration system into the output data format.

20. The system according to claim 12 wherein the communications interface includes a sender for removing executable code from an intermediate form of the outgoing data message to yield an output data format devoid of the executable code for transfer to the business-to-business integration system.

21. The system according to claim 12 further comprising:
   a transaction queue in communication with the reader, the transaction queue storing at least a header of the outgoing data message in a transaction queue in accordance with a priority scheme, the sender sending the outgoing data message in accordance with the priority scheme.
FIG. 3

CREATE A TRANSACTION, WHICH MAY BE EXPRESSED AS A TRANSACTIONAL DOCUMENT CONTAINING TRANSACTIONAL DATA.


INVOKES A READER TO READ OR EXTRACT REFERENCE DATA FOR THE TRANSACTION FROM A DATABASE UPON DETECTION OF THE TRIGGERING EVENT.

FORM AN OUTGOING DATA MESSAGE FROM THE TRANSACTIONAL DATA PLUS THE REFERENCE DATA UPON THE DETECTION OF THE TRIGGERING EVENT. THE OUTGOING DATA MESSAGE IS REPRESENTATIVE OF THE TRANSACTION.

STORE AT LEAST A HEADER OF THE OUTGOING DATA MESSAGE IN A TRANSACTION QUEUE IN ACCORDANCE WITH A PRIORITY SCHEME.

INVOKES THE SENDER TO SEND THE OUTGOING DATA MESSAGE IN ACCORDANCE WITH THE PRIORITY SCHEME.
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
   IPC(7) : G06F 17/60, G 06K 5/00, G 06F 17/00
   US CL : 705/39, 235/380, 705/38, 705/17
   According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
   Minimum documentation searched (classification system followed by classification symbols)
   U.S. : 705/39, 235/380, 705/38, 705/17

   Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

   Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
   EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US 6,105,007 A (NORRIS et. al) 15 AUGUST 2000, ABSTRACT; Fig 1; Fig 2; Fig 3; col 2 lines 36-67; col 3 line 1-col 4 line 14; col 4 lines 25-39; col 5 lines 11-22; col 5 line 43-col 6 line 44; col 7 line 11-col 8 line 41; col 8 line 56-col 9 line 30; col 9 line 61-col 10 line 6</td>
<td>1-21</td>
</tr>
</tbody>
</table>

[ ] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

"A" Special categories of cited documents:
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
23 APRIL 2001

Date of mailing of the international search report
16 JUL 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
VINCENT MILLIN
Telephone No. (703) 308-1065

Form PCT/ISA/210 (second sheet) (July 1998)*
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US 5,880,446 A (MORI et al.) 09 MARCH 1999, Fig 1; col 2 line 8-col 3 line 22; Fig 2; Fig 4; Fig 8; Fig 9; Figs 17-23; col 4 lines 35-51; col 4 line 53-col 5 line 12; col 5 lines 14-24; col 5 line 43-col 5 line 61; col 5 line 65-col 6 line 10; col 6 lines 14-41; col 6 line 45-col 8 line 30; col 8 line 66-col 9 line 20; col 9 line 21-col 10 line 29; col 10 line 52-col 13 line 30; col 13 line 37-col 14 line 65; col 15 line 17-col 18 line 33; col 18 line 42-col 21 line 30; col 21 line 41-col 24 line 32</td>
<td>1-21</td>
</tr>
<tr>
<td>Y</td>
<td>US 6,085,168 A (MORI et al) 04 JULY 2000, ABSTRACT; Figs 1-22; Fig 25; Figs 29-31; Figs 47-51; Figs 56-57; Fig 62; Figs 68-69 Fig 75-76; Fig 84-85; Fig 87-88; Fig 90; col 2 lines 21-29; col 2 lines 52-62; col 3 line 21-col 11 line 35; col 15 line 55-col 20 line 33; col 21 line 34-col 23 line 35; col 23 line 57-col 25 line 50; col 25 line 65-col 30 line 25; col 32 line 18-col 33 line 64; col 34 line 9-col 36 line 37; col 37 line 38-col 39 line 6; col 41 line 29-col 46 line 28; col 51 line 20-col 51 line 62; col 54 line 5-col 54 line 43; col 58 line 8-col 58 line 39; col 61 line 39-col 62 line 14; col 68 lines 40-62; col 73 lines 23-38; col 78 line 43-col 79 line 24; col 81 line 8-col 86 line 65</td>
<td>1-21</td>
</tr>
<tr>
<td>Y</td>
<td>WO 00/52555 (STROH et al.) 08 SEPTEMBER 2000, Figs 5-17; Abstract; pages 1-70</td>
<td>1-21</td>
</tr>
</tbody>
</table>